SINGLE IMAGE BASED CROWD COUNTING USING DEEP LEARNING

by

Vishwanath A. Sindagi

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Doctor of Philosophy

Baltimore, Maryland

March, 2021

Abstract

Estimating count and density maps from crowd images has a wide range of applications such as video surveillance, traffic monitoring, public safety and urban planning. In addition, techniques developed for crowd counting can be applied to related tasks in other fields of study such as cell microscopy, vehicle counting and environmental survey. The task of crowd counting and density map estimation from a single image is a difficult problem since it suffers from multiple issues like occlusions, perspective changes, background clutter, non-uniform density, intra-scene and inter-scene variations in scale and perspective. These issues are further exacerbated in highly congested scenes. In order to overcome these challenges, we propose a variety of different deep learning architectures that specifically incorporate various aspects such as global/local context information, attention mechanisms, specialized iterative and multi-level multi-pathway fusion schemes for combining information from multiple layers in a deep network. Through extensive experimentations and evaluations on several crowd counting datasets, we demonstrate that the proposed networks achieve significant improvements over existing approaches.

We also recognize the need for large amounts of data for training the deep

networks and their inability to generalize to new scenes and distributions. To overcome this challenge, we propose novel semi-supervised and weakly-supervised crowd counting techniques that effectively leverage large amounts of unlabeled/weakly-labeled data. In addition to developing techniques with ability to learn from limited labeled data, we also introduce a new large-scale crowd counting dataset which can be used to train considerably larger networks. The proposed data consists of 4,372 high resolution images with 1.51 million annotations. We made explicit efforts to ensure that the images are collected under a variety of diverse scenarios and environmental conditions. The dataset provides a richer set of annotations like dots, approximate bounding boxes, blur levels, etc.

Primary Reader and Advisor: Prof. Vishal Patel

Secondary Reader: Prof. Rama Chellappa

Acknowledgments

I am forever indebted to the numerous people who supported me throughout my doctoral studies.

First and foremost, I am grateful for the support of my advisor, Professor Vishal Patel, for his patience, kindness, and encouragement. He has guided me and encouraged me to carry on through these years and has contributed to this thesis with a major impact. He has been extremely kind to offer support both on professional and personal fronts.

I thank Professors Rama Chellappa and Alan Yuille for serving on my dissertation and GBO committees, Professors Gregory Hager, Wei Shen and Shinji Watanabe for serving on my GBO committee, and the many wonderful professors who made my education possible.

I would like to thank all my PhD colleagues and collaborators, especially, He Zhang, Poojan, Pramuditha and Rajeev, with whom I have shared moments filled with deep anxiety and excitement at the same time. I believe their presence was very critical in a process that is often felt as enormously solitary. Special thanks to He Zhang, my first collaborator, from whom I learned a lot. A warm word for my colleague and great friend Poojan, who always had the time to discuss and ideate with me, and with whom I had some of

the best coffee breaks. Also, many thanks to Rajeev for his patience through many of the successful collaborations we had together. I am also fortunate to have collaborated with Jose, Vibashan and Deepti, whose enthusiasm always inspired me to work harder. I thank them for all the fun and rich conversations, and most importantly, their valued friendship.

I am eternally grateful to my parents, for the love and support they have bestowed on me through out. To my brother, Manjunath, who encouraged me to join the PhD program, thank you for believing in me. To my sister, Shruthi, thank you for being a strong pillar of support and helping me through many challenging moments.

To my late mother-in-law, who would have been prouder today, thank you for your trust! To my parents-in-law and sister-in-law, Vinuta, who have been whole heartedly supportive of all my endeavours, I am greatly indebted. Many thanks to my brothers-in-law, Praveen and Ankush, and my sister-in-law, Anita, for all their love and support through out this journey.

Some special words of gratitude go to my friends who have always been a major source of support when things would get a bit discouraging: Harish, Raja, Pradeep, Sarvesh, Shiv, Anita, Neelufer, Naveena, Yogesh and Kanna. Thanks guys for always being there for me.

Lastly and most importantly, I am deeply grateful to my wife, Supriya, who has been a bedrock of support since we have known each other. I am greatly indebted to her for supporting all my endeavours and being an integral part of my dreams while sacrificing her own aspirations. This journey would have been impossible without her!

Dedications

This thesis is dedicated to:

My Father - Appasaheb Sindagi,

My Wife - Supriya, and

My Daughter - Mishka.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

With ubiquitous usage of surveillance cameras and advances in computer vision, crowd scene analysis [12, 13] has gained a lot of interest in the recent years. In this thesis, we focus on the task of estimating crowd count and high-quality density maps which has wide applications in video surveillance [14, 15], traffic monitoring, public safety, urban planning [13], scene understanding and flow monitoring. Also, the methods developed for crowd counting can be extended to counting tasks in other fields such as cell microscopy [16, 17, 18, 19], vehicle counting [20, 21, 22, 23, 24], environmental survey [25, 13], etc.

Crowd analysis is an inherently inter-disciplinary research topic with researchers from different communities (such as sociology [26, 27], psychology [28], physics [29, 30], biology [31, 32], computer vision and public safety) have addressed the issue from different viewpoints. Crowd analysis has a variety of critical applications of inter-disciplinarian nature:

Safety monitoring: The widespread usage of video surveillance cameras for

security and safety purposes in places such as sports stadiums, tourist spots, shopping malls and airports has enabled easier monitoring of crowd in such scenarios. However, traditional surveillance algorithms may break down as they are unable to process high density crowds due to limitations in their design. In such scenarios, we can leverage the results of algorithms specially designed for crowd analysis related tasks such as behavior analysis [33, 34], congestion analysis [35, 36], anomaly detection [37, 38] and event detection [39].

Disaster management: Many scenarios involving crowd gatherings such as sports events, music concerts, public demonstrations and political rallies face the risk of crowd related disasters such as stampedes which can be life threatening. In such cases, crowd analysis can be used as an effective tool for early overcrowding detection and appropriate management of crowd, hence, eventual aversion of any disaster [40, 41].

Design of public spaces: Crowd analysis on existing public spots such as airport terminals, train stations, shopping malls and other public buildings [42, 43] can reveal important design shortcomings from crowd safety and convenience point of view. These studies can be used for design of public spaces that are optimized for better safety and crowd movement [44, 45].

Intelligence gathering and analysis: Crowd counting techniques can be used to gather intelligence for further analysis and inference. For instance, in retail sector, crowd counting can be used to gauge people's interest in a product in a store and this information can be used for appropriate product placement [46, 47]. Similarly, crowd counting can be used to measure queue lengths

to optimize staff numbers at different times of the day. Furthermore, crowd counting can be used to analyze pedestrian flow at signals at different times of the day and this information can be used for optimizing signal-wait times [48].

Virtual environments: Crowd analysis methods can be used to understand the underlying phenomenon thereby enabling us to establish mathematical models that can provide accurate simulations. These mathematical models can be further used for simulation of crowd phenomena for various applications such as computer games, inserting visual effects in film scenes and designing evacuation plans [49, 50].

Forensic search: Crowd analysis can be used to search for suspects and victims in events such as bombing, shooting or accidents in large gatherings. Traditional face detection and recognition algorithms can be speeded up using crowd analysis techniques which are more adept at handling such scenarios [51, 52].

These variety of applications has motivated researchers across various fields to develop sophisticated methods for crowd analysis and related tasks such as counting [53, 54, 19, 6, 55, 56, 57, 6], density estimation [18, 58, 1, 59, 60, 16, 61], segmentation [62, 63], behaviour analysis [64, 65, 66, 67, 35, 68], tracking [69, 70], scene understanding [71, 67] and anomaly detection [72, 37]. Among these, crowd counting and density estimation are a set of fundamental tasks and they form basic building blocks for various other applications discussed earlier. Additionally, methods developed for crowd counting can be easily extended to counting tasks in other fields such as cell microscopy

[16, 17, 18, 19], vehicle counting [20], environmental survey [25, 13], etc.

(a) (b)

(c) (d)

Figure 1.1: Illustration of various crowded scenes and the associated challenges. (a) Parade (b) Musical concert (c) Public demonstration (d) Sports stadium. High clutter, overlapping of subjects, variation in scale and perspective can be observed across images.

1.2 Challenges in Crowd Analysis and Counting

Like any other computer vision problem, crowd analysis comes with many challenges such as occlusions, high clutter, non-uniform distribution of people,

non-uniform illumination, intra-scene and inter-scene variations in appearance, scale and perspective making the problem extremely difficult. Figure 1.1 illustrates some of the most important challenges plaguing the crowd counting research community.

In addition to these challenges, the need for significantly large amounts of data for training the deep networks and their inability to generalize to new scenes and distributions further exacerbates these issues since there is an additional burden of collecting and annotating large number of crowd images. However, point-wise annotations are required to learn an effective deep network, and collecting these type of annotations are prohibitively expensive.

1.3 Contributions

In this thesis, we work towards addressing the various problems plaguing the crowd counting research community. These issues can be broadly categorized into (i) task related problems, and (ii) data related problems.

To address the task related challenges such as scale variations, lack of scale annotations, occlusion, etc, we propose a variety of different deep learning architectures that specifically incorporate various aspects such as global/local context information, attention mechanisms, specialized iterative and multi-level multi-pathway fusion schemes for combining information from multiple layers in a deep network. Through extensive experimentations and evaluations on several crowd counting datasets, we demonstrate that the proposed networks achieve significant improvements over existing approaches.

To address the data related problems like the need for large amounts of labeled samples and change in distribution, we propose novel semi-supervised and weakly-supervised crowd counting techniques that effectively leverage large amounts of unlabeled/weakly-labeled data. In addition to developing techniques with ability to learn form limited labeled data, we also introduce a new large-scale crowd counting dataset which can be used to train considerably larger networks.

1.4 Outline

The rest of this thesis is organized into the following chapters:

In Chapter 2, we extensively discuss all the prior works related to crowd counting. First, we present an overview of the broad category of approaches like detection-based methods, regression-based methods and density estimation-based methods. This is followed by an extensive review of traditional approaches and the more recent Convolutional Neural Network (CNN)-based approaches.

In Chapters 3 and 4, we discuss how context information at local and global levels can be incorporated into deep networks for effectively tackling the problem of large variations in scale.

In Chapters 5 and 6, we discuss the methods that we designed to leverage different attention mechanisms for addressing the problems of background clutter and variations in scale.

In Chapters 7 and 8, we discuss different fusion methods that can effectively combine information from different layers in a deep network for

achieving scale-robust solutions. Specifically, we explored two different types of fusion schemes involving multi-level multi-pathway merging and residual based iterative aggregation.

In Chapters 9 and 10, we develop methods to tackle the problem of lack of sufficient labeled data. Specifically, we propose novel weakly-supervised and semi-supervised counting approaches. The weakly supervised approach relies on weak labels like density levels of the crowd image and leverages class activation mapping to generate pixel-wise supervision for the weakly-labeled data. The semi-supervised technique consists of a Gaussian Process-based iterative learning mechanism that involves estimation of pseudo-ground truth for the unlabeled data, which is then used as supervision for training the network.

In Chapter 11, we discuss the details of the large-scale crowd counting dataset collection efforts. Specifically, we describe the need for a new dataset and the various aspects considered while collecting and annotating the samples. In addition, we present the details of the benchmarking experiments conducted to evaluate state-of-the-art approaches on the new dataset.

In Chapter 12, we discuss an application that effectively leverages crowd density maps to detect tiny faces in large crowded images.

Lastly, In Chapter 13, we conclude the thesis and present an outline for the future work.

Chapter 2

Background

Various approaches have been proposed to tackle the problem of crowd counting in images [6, 58, 18, 59, 1] and videos [73, 57, 69, 74]. Loy *et al.* [75] broadly classified traditional crowd counting methods based on the approach into the following categories: (1) Detection-based approaches, (2) Regression-based approaches, and (3) Density estimation-based approaches.

Since the focus of this work is on CNN-based approaches, in this section, we briefly review the detection and regression-based approaches using hand-crafted features for the sake of completeness. In addition, we present a review of the recent traditional methods [6, 18, 60, 16, 76] that have not been analyzed in earlier surveys.

2.1 Detection-based approaches

Most of the initial research was focussed on detection style framework, where a sliding window detector is used to detect people in the scene [77] and this information is used to count the number of people [78]. Detection is usually

performed either in the monolithic style or parts-based detection. Monolithic detection approaches [79, 80, 81, 82] typically are traditional pedestrian detection methods which train a classifier using features (such as Haar wavelets [83], histogram oriented gradients [79], edgelet [84] and shapelet [85]) extracted from a full body. Various learning approaches such as Support Vector Machines, boosting [86] and random forest [87] have been used with varying degree of success. Though successful in low density crowd scenes, these methods are adversely affected by the presence of high density crowds. Researchers have attempted to address this issue by adopting part-based detection methods [88, 89, 90], where one constructs boosted classifiers for specific body parts such as the head and shoulder to estimate the people counts in a designated area [78]. In another approach using shape learning, Zhao et al. [91] modelled humans using 3D shapes composed of ellipsoids, and employed a stochastic process to estimate the number and shape configuration that best explains a given foreground mask in a scene. Ge and Collins [57] further extended the idea by using flexible and practical shape models.

2.2 Regression-based approaches

Though parts-based and shape-based detectors were used to mitigate the issues of occlusion, these methods were not successful in the presence of extremely dense crowds and high background clutter. To overcome these issues, researchers attempted to count by regression where they learn a mapping between features extracted from local image patches to their counts [54, 92, 19]. By counting using regression, these methods avoid dependency on learning

detectors which is a relatively complex task. These methods have two major components: low-level feature extraction and regression modelling. A variety of features such as foreground features, edge features, texture and gradient features have been used for encoding low-level information. Foreground features are extracted from foreground segments in a video using standard background subtraction techniques. Blob-based holistic features such as area, perimeter, perimeter-area ration, etc. have demonstrated encouraging results [53, 19, 92]. While these methods capture global properties of the scene, local features such as edges and texture/gradient features such as local binary pattern (LBP), histogram oriented gradients (HOG), gray level co-occurrence matrices (GLCM) have been used to further improve the results. Once these global and local features are extracted, different regression techniques such as linear regression [93], piecewise linear regression [53], ridge regression [19], Gaussian process regression and neural network [94] are used to learn a mapping from low-level feature to the crowd count.

In a recent approach, Idrees *et al.* [6] identified that no single feature or detection method is reliable enough to provide sufficient information for accurate counting in the presence of high density crowds due to various reasons such as low resolution, severe occlusion, foreshortening and perspective. Additionally, they observed that there exists a spatial relationship that can be used to constrain the count estimates in neighboring local regions. With these observations in mind, they proposed to extract features using different methods that capture different information. By treating densely packed crowds of individuals as irregular and non-homogeneous texture, they employed

Fourier analysis along with head detections and SIFT interest-point based counting in local neighborhoods. The count estimates from this localized multi-scale analysis are then aggregated subject to global consistency constraints. The three sources, i.e., Fourier, interest points and head detection are then combined with their respective confidences and counts at localized patches are computed independently. These local counts are then globally constrained in a multi-scale Markov Random Field (MRF) framework to get an estimate of count for the entire image. The authors also introduced an annotated dataset (UCF_CC_50) of 50 images containing 64000 humans.

Chen *et al.* [58] introduced a novel cumulative attribute concept for learning a regression model when only sparse and imbalanced data are available. Considering that the challenges of inconsistent features along with sparse and imbalanced (encountered during learning a regression function) are related, cumulative attribute-based representation for learning a regression model is proposed. Specifically, features extracted from sparse and imbalanced image samples are mapped onto a cumulative attribute space. The method is based on the notion of discriminative attributes used for addressing sparse training data. This method is inherently capable of handling imbalanced data.

2.3 Density estimation-based approaches

While the earlier methods were successful in addressing the issues of occlusion and clutter, most of them ignored important spatial information as they were regressing on the global count. In contrast, Lempitsky *et al.* [18] proposed to learn a linear mapping between local patch features and corresponding

object density maps, thereby incorporating spatial information in the learning process. In doing so, they avoided the hard task of learning to detect and localize individual object instances by introducing a new approach of estimating image density whose integral over any region in the density map gives the count of objects within that region. The problem of learning density maps is formulated as a minimization of a regularized risk quadratic cost function. A new loss function appropriate for learning density maps is introduced. The entire problem is posed as a convex optimization task which they solve using cutting-plane optimization.

Observing that it is difficult to learn a linear mapping, Pham *et al.* [60] proposed to learn a non-linear mapping between local patch features and density maps. They used random forest regression from multiple image patches to vote for densities of multiple target objects to learn a non-linear mapping. In addition, they tackled the problem of large variation in appearance and shape between crowded image patches and non-crowded ones by proposing a crowdedness prior and they trained two different forests corresponding to this prior. Furthermore, they were able to successfully speed up the estimation process for real-time performance by proposing an effective forest reduction that uses permutation of decision trees. Apart from achieving real-time performance, another advantage of their method is that it requires relatively less memory to build and store the forest.

Similar to the above approach, Wang and Zou [16] identified that though existing methods are effective, they were inefficient from computational complexity point of view. To this effect, they proposed a fast method for density

estimation based on subspace learning. Instead of learning a mapping between dense features and their corresponding density maps, they learned to compute the embedding of each subspace formed by image patches. Essentially, they exploited the relationship between images and their corresponding density maps in the respective feature spaces. The feature space of image patches are clustered and examples of each subspace are collected to learn its embedding. Their assumption that local image patches and their corresponding density maps share similar local geometry enables them to learn locally linear embedding using which the density map of an image patch can be estimated by preserving the geometry. Since, implementing locally linear embedding (LLE) is time-consuming, they divided the feature spaces of image patches and their counterpart density maps into subspaces, and computed the embedding of each subspace formed by image patches. The density map of input patch is then estimated by simple classification and mapping with the corresponding embedding matrix.

In a more recent approach, Xu and Qiu [76] observed that the existing crowd density estimation methods used a smaller set of features thereby limiting their ability to perform better. Inspired by the ability of high-dimensional features in other domains such as face recognition, they proposed to boost the performances of crowd density estimation by using a much extensive and richer set of features. However, since the regression techniques used by earlier methods (based on Gaussian process regression or Ridge regression) are computationally complex and are unable to process very high-dimensional features, they used random forest as the regression model whose tree structure

is intrinsically fast and scalable. Unlike traditional approaches to random forest construction, they embedded random projection in the tree nodes to combat the curse of dimensionality and to introduce randomness in the tree construction.

2.4 CNN-based methods

The success of CNNs in numerous computer vision tasks has inspired researchers to exploit their abilities for learning non-linear functions from crowd images to their corresponding density maps or corresponding counts. A variety of CNN-based methods have been proposed in the literature. We broadly categorize these methods based on property of the networks and training approach. Based on the property of the networks, we classify the approaches into the following categories:

- **Basic CNNs**: Approaches that involve basic CNN layers in their networks fall into this category. These methods are amongst initial deep learning approaches for crowd counting and density estimation.
- **Scale-aware models**: The basic CNN-based approaches evolved into more sophisticated models that were robust to variations in scale. This robustness is achieved through different techniques such as multi-column or multi-resolution architectures.
- **Context-aware models**: Another set of approaches attempted to incorporate local and global contextual information present in the image into the CNN framework for achieving lower estimation errors.

- **Multi-task frameworks**: Motivated by the success of multi-task learning for various computer vision tasks, various approaches have been developed to combine crowd counting and estimation along with other tasks such as foreground-background subtraction and crowd velocity estimation.

In an yet another categorization, we classify the CNN-based approaches based on the inference methodology into the following two categories:

- **Patch-based inference**: In this approach, the CNNs are trained using patches cropped from the input images. Different methods use different crop sizes. During the prediction phase, a sliding window is run over the test image and predictions are obtained for each window and finally aggregated to obtain total count in the image.
- **Whole image-based inference**: Methods in this category perform a whole-image based inference. These methods avoid computationally expensive sliding windows.

Table 2.1 presents a categorization of various CNN-based crowd counting methods based on their network property and inference process.

Next, we review various CNN-based crowd counting and density estimation methods along with their merits and drawbacks.

Wang *et al.* [96] and Fu *et al.* [95] were among the first ones to apply CNNs for the task of crowd density estimation. Wang *et al.* proposed an end-to-end deep CNN regression model for counting people from images in extremely dense crowds.

Table 2.1: Categorization of existing CNN-based approaches.

	Category	
Method	Network property	Inference process
Fu *et al.* [95]	Basic	Patch-based
Wang *et al.* [96]	Basic	Patch-based
Zhang *et al.* [59]	Multi-task	Patch-based
Boominathan *et al.* [61]	Scale-aware	Patch-based
Zhang *et al.* [1]	Scale-aware	Whole image-based
Walach and Wolf [17]	Basic	Patch-based
Onoro *et al.* [20]	Scale-aware	Patch-based
Shang *et al.* [56]	Context-aware	Whole image-based
Sheng *et al.* [97]	Context-aware	Whole image-based
Kumagai *et al.* [98]	Scale-aware	Patch-based
Marsden *et al.* [99]	Scale-aware	Whole image-based
Mundhenk *et al.* [100]	Basic	Patch-based
Artetta *et al.* [101]	Multi-task	Patch-based
Zhao *et al.* [102]	Multi-task	Patch-based
Sindagi *et al.* [103]	Multi-task	Whole image-based
Sam *et al.* [104]	Scale-aware	Patch-based
Kang *et al.* [102]	Basic	Patch-based

Zhang *et al.* [59] analyzed existing methods to identify that their performance reduces drastically when applied to a new scene that is different from the training dataset. To overcome this issue, they proposed to learn a mapping from images to crowd counts and to adapt this mapping to new target scenes for cross-scene counting. Additionally, they introduced a new dataset for the purpose of evaluating cross-scene crowd counting. The network is evaluated for cross-scene crowd counting as well as single scene crowd counting and superior results are demonstrated for both scenarios.

Inspired by the success of cross-scene crowd counting [59], Walach and Wolf [17] performed layered boosting and selective sampling. This layered boosting approach is based on the notion of Gradient Boosting Machines (GBM) [105] which are a subset of powerful ensemble techniques. In an effort

to capture semantic information in the image, Boominathan *et al.* [61] combined deep and shallow fully convolutional networks to predict the density map for a given crowd image.

In another approach, Zhang *et al.* [1] proposed a multi-column based architecture (MCNN) for images with arbitrary crowd density and arbitrary perspective. Inspired by the success of multi-column networks for image recognition [106], the proposed method ensures robustness to large variation in object scales by constructing a network that comprises of three columns corresponding to filters with receptive fields of different sizes (large, medium, small). Finally, considering that existing crowd counting datasets do not cater to all the challenging situations encountered in real world scenarios, a new ShanghaiTech crowd datasets is constructed. This new dataset includes 1,198 images with about 330,000 annotated heads.

Similar to the above approach, Onoro and Sastre [20] developed a scale aware counting model called Hydra CNN that is able to estimate object densities in a variety of crowded scenarios without any explicit geometric information of the scene. The network consists of 3 heads and a body with each head learning features for a particular scale. Each head of the Hydra-CNN is constructed using the CCNN model whose outputs are concatenated and fed to the body. The body consists of a set of two fully-connected layers followed by a rectified linear unit (ReLu), a dropout layer and a final fully connected layer to estimate the object density map. While the different heads extract image descriptors at different scales, the body learns a high-dimensional representation that fuses the multi-scale information provided by the heads.

Instead of training all regressors of a multi-column network [1] on all the input patches, Sam *et al.* [104] argue that better performance is obtained by training regressors with a particular set of training patches by leveraging variation of crowd density within an image. To this end, they proposed a switching CNN that cleverly selects an optimal regressor suited for a particular input patch.

In another approach, Cao *et al.* [107] proposed a encoder-decoder network with scale aggregation modules.

In contrast to these methods that emphasize on specifically addressing large-scale variations in head sizes, the most recent methods ([108] ,[109], [110], [111], [112]) have focused on other properties of the problem. For instance, Babu *et al.* [108] proposed a mechanism to incrementally increase the network capacity conditioned on the dataset. Shen *et al.* [109] overcame the issue of blurred density maps by utilizing adversarial loss. In a more recent approach, Ranjan *et al.* [112] proposed a two-branch network to estimate density map in a cascaded manner. Shi *et al.* [110] employed deep negative correlation based learning for more generalizable features. Liu *et al.* [111] used unlabeled data for counting by proposing a new framework that involves learning to rank.

Recent approaches like [113, 114, 115, 116, 117, 118] have aimed at incorporating various forms of related information like attention [113], semantic priors [114], segmentation [115], inverse attention [116], and hierarchical attention [117] respectively into the network. Other techniques such as [119, 120, 121, 122, 123] leverage features from different layers of the network using different techniques like trellis style encoder decoder [119], explicitly

considering perspective [120], context information [121], adaptive density map generation [123] and multiple views [122]. More recently, Sam *et al.* [124] introduced a detection framework for densely crowded scenarios where the network is trained using estimated bounding-boxes. Ma *et al.* [125] proposed a novel Bayesian loss function for training counting networks, which involves supervision on the count expectation at each annotated point. While most of the existing approaches are focused on counting in 2D plane, Zhang *et al.* [126] propose to solve the multi-view crowd counting task through 3D feature fusion with 3D scene-level density maps.

For a comprehensive study on various crowd counting techniques, the reader is referred to detailed surveys like [127, 128].

2.5 Crowd Datasets

Crowd counting datasets have evolved over time with respect to a number of factors such as size, crowd densities, image resolution, and diversity. UCSD [53] is among one of the early datasets proposed for counting and it contains 2000 video frames of low resolution with 49,885 annotations. The video frames are collected from a single frame and typically contain low density crowds. Zhang *et al.* [59] addressed the limitations of UCSD dataset by introducing the WorldExpo dataset that contains 108 videos with a total of 3,980 frames belonging to 5 different scenes. While the UCSD and WorldExpo datasets contain only low/low-medium densities, Idrees *et al.* [6] proposed the UCF_CROWD_50 dataset specifically for very high density crowd scenarios. However, the dataset consists of only 50 images rendering it impractical for

training deep networks. Zhang *et al.* [1] introduced the ShanghaiTech dataset which has better diversity in terms of scenes and density levels as compared to earlier datasets. The dataset is split into two parts: Part A (containing high density crowd images) and Part B (containing low density crowd images). The entire dataset contains 1,198 images with 330,165 annotations. Recently, Idrees *et al.* [2] proposed a new large-scale crowd dataset containing 1,535 high density images images with a total of 1.25 million annotations. Wang *et al.* [4] introduced a synthetic crowd counting dataset that is based on GTA V electronic game. The dataset consists of 15,212 crowd images under a diverse set of scenes. In addition, they proposed a SSIM based CycleGAN [129] for adapting the network trained on synthetic images to real world images. Most recently, Wang *et al.* [130] released a large-scale crowd counting dataset (NWPUCrowd) consisting of 5,109 images with 2.13 million annotations.

2.6 Evaluation Metrics

For the purpose of evaluation, the standard metrics used by many existing methods for crowd counting were used. These metrics are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'|, \tag{2.1}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|y_i - y_i'|^2} \tag{2.2}$$

where MAE is mean absolute error, MSE is mean squared error, N is number of test samples, y_i is ground truth count and y_i' is estimated count corresponding to the i^{th} sample.

Chapter 3

Cascaded Multi Task Learning based Counting Network

One of the many challenges faced by researchers working on crowd counting is the issue of large variations in scale and appearance of the objects that occurs due to severe perspective distortion of the scene. Many methods have been developed that incorporate scale information into the learning process using different methods. Some of the early methods relied on multi-source and hand-crafted representations and catered only to low density crowded scenes [6]. These methods are rendered ineffective in high density crowds and the results are far from optimal. Inspired by the success of Convolutional Neural Networks (CNNs) for various computer vision tasks, many CNN-based methods have been developed to address the problem of crowd counting [61, 131, 59]. Considering scale issue as a limiting factor to achieve better accuracies, certain CNN-based methods specifically cater to the issue of scale changes via multi-column or multi-resolution network [1, 20, 56]. Though these methods demonstrated robustness to scale changes, they are still restricted to the scales that are used during training and hence are limited

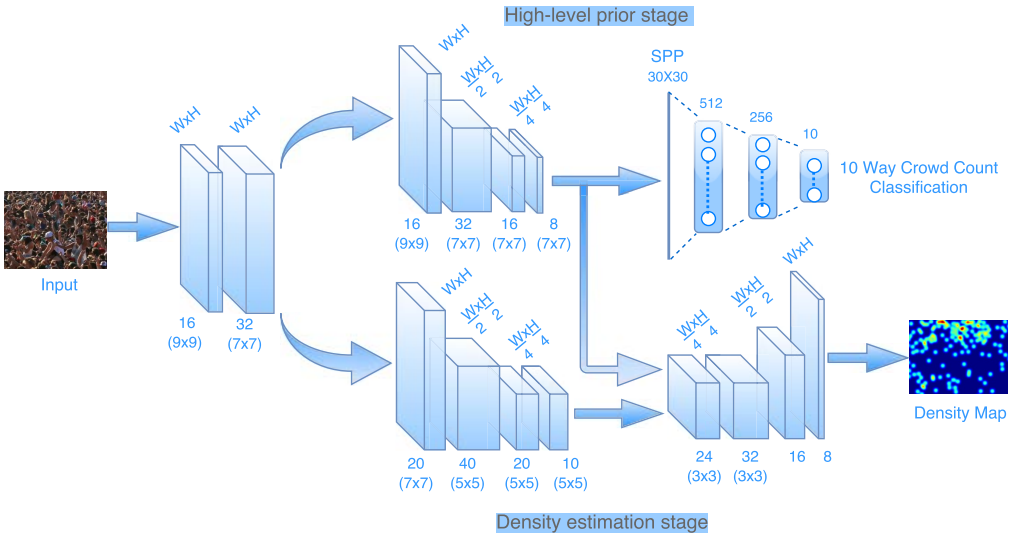


Figure 3.1: Overview of the proposed cascaded architecture for jointly learning high-level prior and density estimation.

in their capacity to learn well-generalized models.

The aim of this work is to learn models that cater to a wide variety of density levels present in the dataset by incorporating a high-level prior into the network. The high-level prior learns to classify the count into various groups whose class labels are based on the number of people present in the image. By exploiting count labels, the high-level prior is able to estimate coarse count of people in the entire image irrespective of scale variations thereby enabling the network to learn more discriminative global features. The high-level prior is jointly learned along with density map estimation using a cascade of CNN networks.

3.1 Proposed method

Inspired by the success of cascaded convolutional networks for related multiple tasks [132, 133, 134], we propose to learn two related sub-tasks: crowd count classification (which we call as high-level prior) and density map estimation in a cascaded fashion as shown in Figure 3.1. The network takes an image of arbitrary size, and outputs crowd density map. The cascaded network has two stages corresponding to the two sub-tasks, with the first stage learning high-level prior and the second stage preforming density map estimation. Both stages share a set of convolutional features. The first stage consists of a set of convolutional layers and spatial pyramid pooling to handle arbitrarily sized images followed by a set of fully connected layers. The second stage consists of a set of convolutional layers followed by fractionally-strided convolutional layers for upsampling the previous layer's output to account for the loss of details due to earlier pooling layers. Two different set of loss layers are used at the end of the two stages, however, the loss of the second layer is dependent on the output of the earlier stage. The following sub-sections discuss the details of all the components of the proposed network.

3.1.1 Shared convolutional layers

The initial shared network consists of 2 convolutional layers with a Parametric Rectified Linear Unit (PReLU) activation function after every layer. The first convolutional layer has 16 feature maps with a filter size of 9×9 and the second convolutional layer has 32 feature maps with a filter size of 7×7. The feature maps generated by this shallow network are shared by the two stages:

high-level prior stage and density estimation stage.

3.1.2 High-level prior stage

Classifying the crowd into several groups is an easier problem as compared to directly performing classification or regression for the whole count range which requires a larger amount of training data. Hence, we quantize the crowd count into ten groups and learn a crowd count group classifier which also performs the task of incorporating high-level prior into the network. The high-level prior stage takes feature maps from the previous shared convolutional layers. This stage consists of 4 convolutional layers with a PReLU activation function after every layer. The first two layers are followed by max pooling layers with a stride of 2. At the end, the high-level prior stage consists of three fully connected (FC) layers with a PReLU activation function after every layer. The first FC layer consists of 512 neurons whereas the second FC layer consists of 256 neurons. The final layer consists of a set of 10 neurons followed by a sigmoid layer, indicating the count class of the input image. To enable the use of arbitrarily sized images for training, Spatial Pyramid Pooling (SPP) [135] is employed as it eliminates the fixed size constraint of deep networks which contain fully connected layers. The SPP layer is inserted after the last convolutional layer. The SPP layer aggregates features from the convolutional layers to produce fixed size outputs and can be fed to the fully connected layers. Cross-entropy error is used as the loss layer for this stage.

3.1.3 Density estimation

The feature maps obtained from the shared layers are processed by an another CNN network that consists of 4 convolutional layers with a PReLU activation function after every layer. The first two layers are followed by max pooling layers with a stride of 2, due to which the output of CNN layers is down-sampled by a factor of 4. The first convolutional layer has 20 feature maps with a filter size of 7×7, the second convolutional layer has 40 feature maps with a filter size of 5×5, the third layer has 20 feature maps with a filter size of 5×5 and the fourth layer has 10 feature maps with a filter size of 5×5. The output of this network is combined with that of the last convolutional layer of high-level prior stage using a set of 2 convolutional and 2 fractionally strided convolutional layers. The first two convolutional layers have a filter size of 3×3 with 24 and 32 feature maps, respectively. These layers are followed by 2 sets of fractionally strided convolutional layers with 16 and 18 feature maps, respectively. In addition to integrating high-level prior from an earlier stage, the fractionally strided convolutions learn to upsample the feature maps to the original input size thereby restoring the details lost due to earlier max-pooling layers. The use of these layers results in upsampling of the CNN output by a factor of 4, thus enabling us to regress on full resolution density maps. Standard pixel-wise Euclidean loss is used as the loss layer for this stage. Note that this loss depends on intermediate output of the earlier cascade, thereby enforcing a causal relationship between count classification and density estimation.

3.1.4 Objective function

The cross-entropy loss function for the high-level prior stage is defined as follows:

$$L_c = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}[(y^i = j)F_c(X_i, \Theta)], \tag{3.1}$$

where N is number of training samples, Θ is a set of network parameters, X_i is the i^{th} training sample, $F_c(X_i, \Theta)$ is the classification output, y^i is the ground truth class and M is the total number of classes.

The loss function for the density estimation stage is defined as:

$$L_d = \frac{1}{N}\sum_{i=1}^{N} \|F_d(X_i, C_i, \Theta) - D_i\|_2, \tag{3.2}$$

where $F_d(X_i, C_i, \Theta)$ is the estimated density map, D_i is the ground truth density map, and C_i are the feature maps obtained from the last convolutional layer of the high-level prior stage.

The entire cascaded network is trained using the following unified loss function:

$$L = \lambda L_c + L_d, \tag{3.3}$$

where λ is a weighting factor.

This loss function is unlike traditional multi-task learning, because the loss term of the last stage depends on the output of the earlier one.

3.1.5 Training and implementation details

In this section, details of the training procedure are discussed. To create the training dataset, patches of size $1/4^{th}$ the size of original image are cropped from 100 random locations. Other augmentation techniques like horizontal flipping and noise addition are used to create another 200 patches. The random cropping and augmentation resulted in a total of 300 patches per image in the training dataset. Note that the cropping is used only as a data augmentation technique and the resulting patches are of arbitrary sizes.

Several sophisticated methods are proposed in the literature for calculating the ground truth density map [59, 1]. We use a simple method in order to ensure that the improvements achieved are due to the proposed method and are not dependent on the sophisticated methods for calculating the ground truth density maps. Ground truth density map D_i corresponding to the i^{th} training patch is calculated by summing a 2D Gaussian kernel centered at every person's location x_g as defined below:

$$D_i(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma), \tag{3.4}$$

where σ is the scale parameter of the 2D Gaussian kernel and S is the set of all points at which people are located.

The training and evaluation was performed on NVIDIA GTX TITAN-X GPU using Torch framework [136]. λ was set to 0.0001 in (5.3). Adam optimization with a learning rate of 0.00001 and momentum of 0.9 was used to train the model. Additionally, for the classification (high-level prior) stage, to account for the imbalanced datasets, the losses for each class were weighted

based on the number of samples available for that particular class. The training took approximately 6 hours.

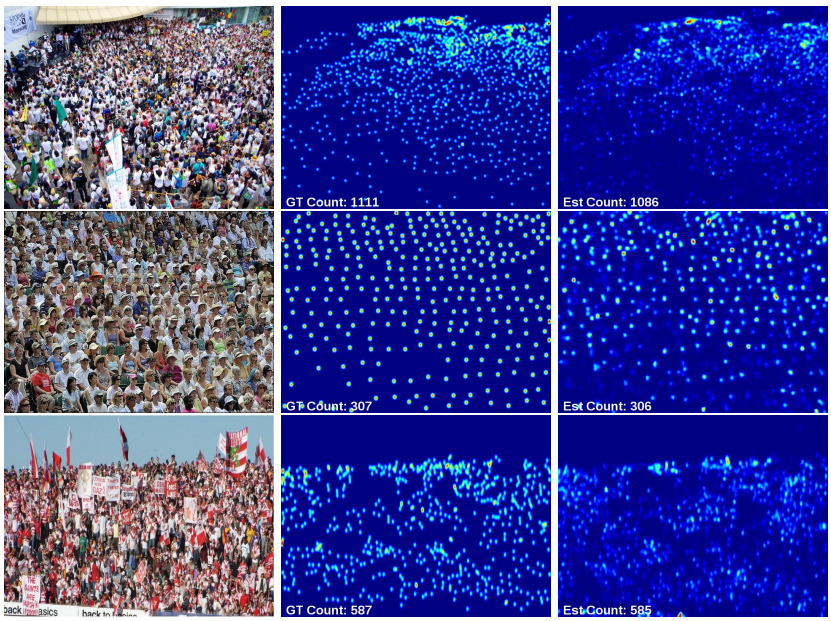
3.2 Experimental results

In this section, we present the experimental details and evaluation results on two publicly available datasets: ShanghaiTech [1] and UCF_CROWD_50 [6].

3.2.1 ShanghaiTech dataset

The results of the proposed method are compared with two recent approaches: Zhang *et al.* [59] and MCNN by Zhang *et al.* [1] (Table 4.2). It can be observed that the proposed method is able to achieve significant improvements without the use of multi-column networks or sophisticated ground truth map generation. Furthermore, to demonstrate the improvements obtained by incorporating high-level prior via cascaded architecture, we evaluated our network without the high-level prior stage (Single stage CNN) on ShanghaiTech dataset. It can also be observed that the cascaded learning of count classification and density estimation reduces the count error by a large margin as compared to the single stage CNN.

Figure 3.2 illustrates the density map results obtained using the proposed method as compared to Zhang *et al.* [1] and single stage CNN. It can be observed that in addition to achieving lower count error, the proposed method results in higher quality density maps due to the use of fractionally strided convolutional layers.



(a) (b) (c)

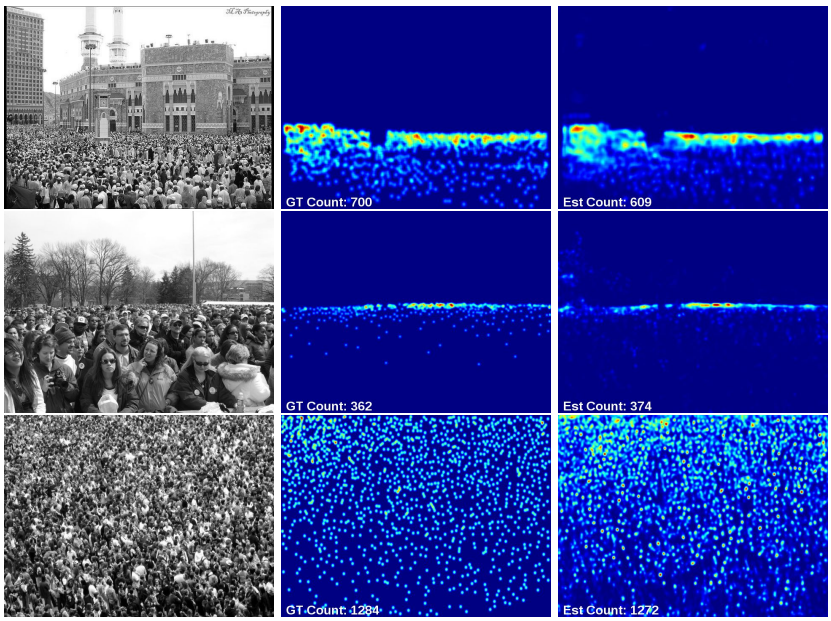
Figure 3.2: Density estimation results using proposed method on ShanghaiTech dataset. (a) Input (b) Ground truth (c) Output.

Table 3.1: Comparison results: Estimation errors on the ShanghaiTech dataset. The proposed method achieves lower error compared to existing approaches involving multi column CNNs and sophisticated density maps.

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Zhang *et al.* [59]	181.8	277.7	32.0	49.8
MCNN [1]	110.2	173.2	26.4	41.3
Single stage CNN	130.4	190.9	29.3	40.5
Proposed method	**101.3**	**152.4**	**20.0**	**31.1**

3.2.2 UCF_CC_50 dataset

Following the standard protocol discussed in [6], a 5-fold cross-validation was performed for evaluating the proposed method. The results are compared with five recent approaches: Idrees *et al.* [6], Zhang *et al.* [59], MCNN [1], Onoro *et al.* [20] and Walach *et al.* [17]. It can be observed from Table 8.6 that our network achieves the lowest MAE and comparable MSE score. Density maps



(a) (b) (c)

Figure 3.3: Density estimation results using proposed method on UCF_CC_50 dataset. (a) Input (b) Ground truth (c) Output.

obtained using the proposed method on sample images from UCF_CC_50 dataset are shown in Figure 3.3.

Table 3.2: Comparison results: Estimation errors on the UCF_CC_50 dataset.

Method	MAE	MSE
Idrees *et al.* [6]	419.5	541.6
Zhang *et al.* [59]	467.0	498.5
MCNN [1]	377.6	509.1
Onoro *et al.* [20]	465.7	371.8
Walach *et al.* [17]	364.4	**341.4**
Proposed method	**322.8**	397.9

3.3 Summary

We presented a multi-task cascaded CNN network for jointly learning crowd count classification and density map estimation. By learning to classify the crowd count into various groups, we are able to incorporate a high-level prior into the network which enables it to learn globally relevant discriminative features thereby accounting for large count variations in the dataset. Additionally, we employed fractionally strided convolutional layers at the end so as to account for the loss of details due to max-pooling layers in the earlier stages there by allowing us to regress on full resolution density maps. The entire cascade was trained in an end-to-end fashion. Extensive experiments performed on challenging datasets and comparison with recent state-of-the-art approaches demonstrated the significant improvements achieved by the proposed method.

Chapter 4

Scale Aware Counting Using Contextual Pyramid CNNs

Existing CNN-based methods using different multi-scale architectures [1, 20, 104] have achieved significant success in addressing some of the above issues, especially in the high-density complex crowded scenes. However, these methods tend to under-estimate or over-estimate count in the presence of high-density and low-density crowd images, respectively. A potential solution is to use contextual information during the learning process. Several recent works for semantic segmentation [137], scene parsing [138] and visual saliency [139] have demonstrated that incorporating contextual information can provide significant improvements in the results. Motivated by their success, we believe that availability of global context shall aid the learning process and help us achieve better count estimation. In addition, existing approaches employ max-pooling layers to achieve minor translation invariance resulting in low-resolution and hence low-quality density maps. Also, to the best of our knowledge, most existing methods concentrate only on the quality of count rather than that of density map. Considering these observations, we propose

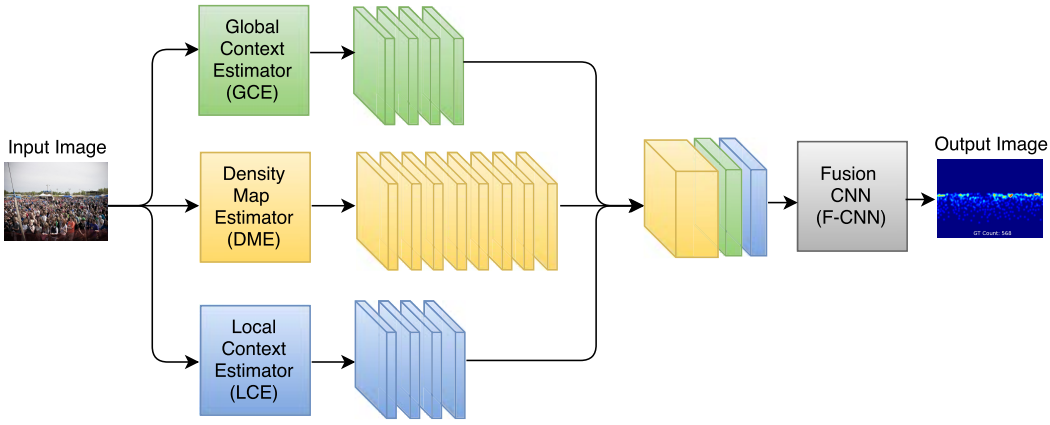


Figure 4.1: Overview of the proposed CP-CNN architecture. The network incorporates global and local context using *GCE* and *LCE* respectively. The context maps are concatenated with the output of *DME* and further processed by *F-CNN* to estimate high-quality density maps.

to incorporate global context into the learning process while improving the quality of density maps.

4.1 Proposed method (CP-CNN)

The proposed CP-CNN method consists of a pyramid of context estimators and a Fusion-CNN as illustrated in Figure 4.1. It consists of four modules: *GCE*, *LCE*, *DME*, and *F-CNN*. *GCE* and *LCE* are CNN-based networks that encode global and local context present in the input image respectively. *DME* is a multi-column CNN that performs the initial task of transforming the input image to high-dimensional feature maps. Finally, *F-CNN* combines contextual information from *GCE* and *LCE* with high-dimensional feature maps from *DME* to produce high-resolution and high-quality density maps. These modules are discussed in detail as follows.

4.1.1 Global Context Estimator (GCE)

Although recent state-of-the-art multi-column or multi-scale methods [1, 20, 17] achieve significant improvements in the task of crowd count estimation, they either underestimate or overestimate counts in high-density and low-density crowd images respectively. We believe it is important to explicilty model context present in the image to reduce the estimation error. To this end, we associate global context with the level of density present in the image by considering the task of learning global context as classifying the input image into five different classes: extremely low-density (ex-lo), low-density (lo), medium-density (med), high-density (hi) and extremely high-density (ex-hi). Note that the number of classes required is dependent on the crowd density variation in the dataset. A dataset containing large variations may require higher number of classes. In our experiments, we obtained significant improvements using five categories of density levels.

In order to learn the classification task, a VGG-16 [140] based network is fine-tuned with the crowd training data. Network used for *GCE* is as shown in Figure 4.2. The convolutional layers from the VGG-16 network are retained, however, the last three fully connected layers are replaced with a different configuration of fully connected layers in order to cater to our task of classification into five categories. Weights of the last two convolutional layers are fine-tuned while keeping the weights fixed for the earlier layers. The use of pre-trained VGG network results in faster convergence as well as better performance in terms of context estimation.

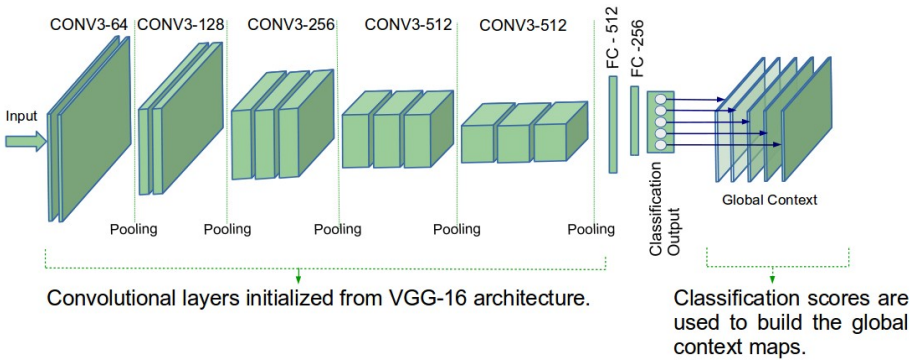


Figure 4.2: Global context estimator based on VGG-16 architecture. The network is trained to classify the input images into various density levels thereby encoding the global context present in the image.

4.1.2 Local Context Estimator (LCE)

Existing methods for crowd density estimation have primarily focussed on achieving lower count errors rather than estimating better quality density maps. After an analysis of these results, we believe that some kind of local contextual information can aid us to achieve better quality maps. To this effect, similar to *GCE*, we propose to learn an image's local context by learning to classify it's local patches into one of the five classes: {ex-lo, lo, med, hi, ex-hi}. The local context is learned by the *LCE* whose architecture shown in Figure 4.3. It is composed of a set of convolutional and max-pooling layers followed by 3 fully connected layers with appropriate drop-out layers after the first two fully connected layers. Every convolutional and fully connected layer is followed by a ReLU layer except for the last fully connected layer which is followed by a sigmoid layer.

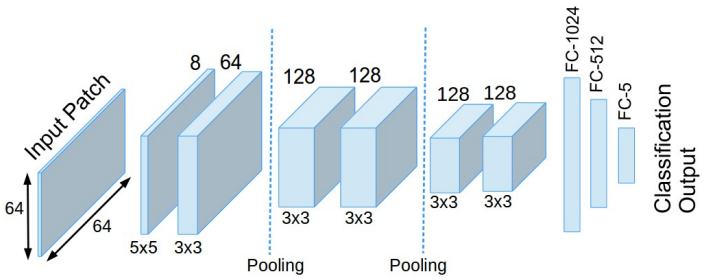


Figure 4.3: Local context estimator: The network is trained to classify local input patches into various density levels thereby encoding the local context present in the image.

4.1.3 Density Map Estimator (DME)

The aim of *DME* is to transform the input image into a set of high-dimensional feature maps which will be concatenated with the contextual information provided by *GCE* and *LCE*. Estimating density maps from high-density crowd images is especially challenging due to the presence of heads with varying sizes in and across images. Previous works on multi-scale [20] or multi-column [1] architectures have demonstrated abilities to handle the presence of considerably large variations in object sizes by achieving significant improvements in such scenarios. Inspired by the success of these methods, we use a multi-column architecture similar to [1]. However, notable differences compared to their work are that our columns are much deeper and have different number of filters and filter sizes that are optimized for lower count estimation error. In addition, the multi-column architecture is used to transform the input into a set of high-dimensional feature map rather than using them directly to estimate the density map. Network details for *DME* are illustrated in Figure 4.4.

It may be argued that since the *DME* has a pyramid of filter sizes, one

may be able to increase the filter sizes and number of columns to address larger variation in scales. However, note that addition of more columns and the filter sizes will have to be decided based on the scale variation present in the dataset, resulting in new network designs that cater to different datasets containing different scale variations. Additionally, deciding the filter sizes will require time consuming experiments. With our network, the design remains consistent across all datasets, as the context estimators can be considered to perform the task of coarse crowd counting.

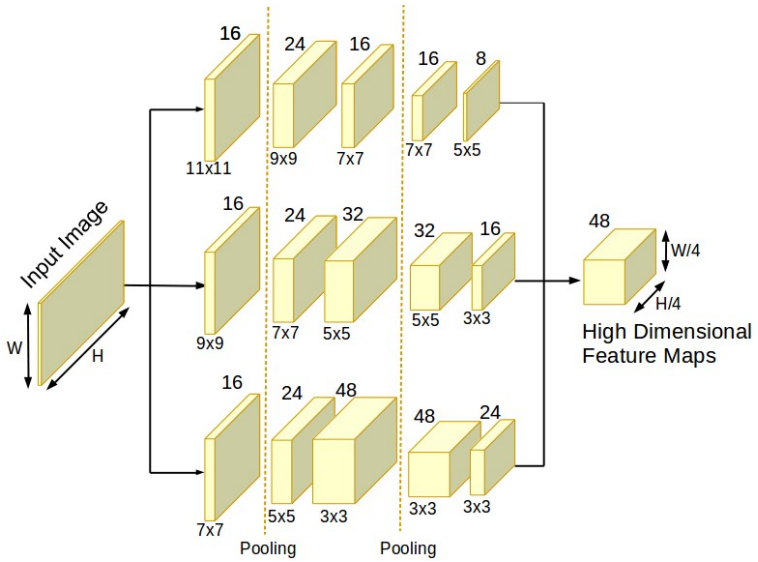


Figure 4.4: Density Map Estimator: Inspired by Zhang *et al.* [1], DME is a multi-column architecture. In contrast to [1], we use slightly deeper columns with different number of filters and filter sizes.

4.1.4 Fusion-CNN (F-CNN)

The contextual information from *GCE* and *LCE* are combined with the high-dimensional feature maps from *DME* using *F-CNN*. The *F-CNN* automatically

learns to incorporate the contextual information estimated by context estimators. The presence of max-pooling layers in the *DME* network (which are essential to achieve translation invariance) results in down-sampled feature maps and loss of details. Since, the aim of this work is to estimate high-resolution and high-quality density maps, *F-CNN* is constructed using a set of convolutional and fractionally-strided convolutional layers. The set of fractionally-strided convolutional layers help us to restore details in the output density maps. The following structure is used for *F-CNN*: *CR(64,9)-CR(32,7)-TR(32)-CR(16,5)-TR(16)-C(1,1)*, where, *C* is convolutional layer, *R* is ReLU layer, *T* is fractionally-strided convolution layer and the first number inside every brace indicates the number of filters while the second number indicates filter size. Every fractionally-strided convolution layer increases the input resolution by a factor of 2, thereby ensuring that the output resolution is the same as that of input.

Once the context estimators are trained, *DME* and *F-CNN* are trained in an end-to-end fashion. Existing methods for crowd density estimation use Euclidean loss to train their networks. It has been widely acknowledged that minimization of L_2 error results in blurred results especially for image reconstruction tasks [141, 142, 143, 144, 145]. Motivated by these observations and the recent success of GANs for overcoming the issues of L2-minimization [141], we attempt to further improve the quality of density maps by minimizing a weighted combination of pixel-wise Euclidean loss and adversarial loss.

The loss for training *F-CNN* and *DME* is defined as follows:

$$L_T = L_E + \lambda_a L_A, \tag{4.1}$$

$$L_E = \frac{1}{WH} \sum_{w=1}^{W} \sum_{h=1}^{H} \|\phi(X^{w,h}) - (Y^{w,h})\|_2, \tag{4.2}$$

$$L_A = -\log(\phi_D(\phi(X)), \tag{4.3}$$

where, L_T is the overall loss, L_E is the pixel-wise Euclidean loss between estimated density map and it's corresponding ground truth, λ_a is a weighting factor, L_A is the adversarial loss, X is the input image of dimensions $W \times H$, Y is the ground truth density map, ϕ is the network consisting of *DME* and *F-CNN* and ϕ_D is the discriminator sub-network for calculating the adversarial loss. Following structure is used for the discriminator sub-network: *CP(64)-CP(128)-M-CP(256)-M-CP(256)-CP(256)-M-C(1)-Sigmoid*, where *C* represents convolutional layer, *P* represents PReLU layer and *M* is max-pooling layer.

4.2 Training and evaluation details

In this section, we discuss details of the training and evaluation procedures.

Training details: Let D be the original training dataset. Patches $1/4^{th}$ the size of original image are cropped from 100 random locations from every image in D. Other augmentation techniques like horizontal flipping and noise addition are used to create another 200 patches. The random cropping and augmentation resulted in a total of 300 patches per image in the training dataset. Let this set of images be called as D_{dme}. Another training set D_{lc}

is formed by cropping patches of size 64×64 from 100 random locations in every training image in D.

GCE is trained using the dataset D_{dme}. The corresponding ground truth categories for each image is determined based on the number of people present in it. Note that the images are resized to 224×224 before feeding them into the VGG-based *GCE* network. The network is then trained using the standard cross-entropy loss. *LCE* is trained using the 64×64 patches in D_{lc}. The ground truth categories of the training patches is determined based on the number of people present in them. The network is then trained using the standard cross-entropy loss.

Next, the *DME* and *F-CNN* networks are trained in an end-to-end fashion using input training images from D_{dme} and their corresponding global and local contexts[1]. The global context (F^i_{gc}) for an input training image X^i is obtained in the following way. First, an empty global context F^i_{gc} of dimension $5 \times W_i/4 \times H_i/4$ is created, where $W_i \times H_i$ is the dimension of X_i. Next, a set of classification scores $y^{i,j}_{gc}(j = 1...5)$ is obtained by feeding X_i to *GCE*. Each feature map in global context $F^{i,j}_{gc}$ is then filled with the corresponding classification score $y^{i,j}_{g}$. The local context (F^i_{lc}) for X^i is obtained in the following way. An empty local context F^i_{lc} of dimension $5 \times W_i \times H_i$ is first created. A sliding window classifier (*LCE*) of size 64×64 is run on X_i to obtain the classification score $y^{i,j,w}_{lc}(j = 1...5)$ where w is the window location. The classification scores $y^{i,j,w}_{lc}$ are used to fill the corresponding window location w in the respective local context map $F^{i,j}_{gc}$. $F^{i,j}_{gc}$ is then resized to a size

[1]Once *GCE* and *LCE* are trained, their weights are frozen.

of $W_i/4 \times H_i/4$. After the context maps are estimated, X_i is fed to *DME* to obtain a high-dimensional feature map F^i_{dme} which is concatenated with F^i_{gc} and F^i_{lc}. These concatenated feature maps are then fed into *F-CNN*. The two CNNs (*DME* and *F-CNN*) are trained in an end-to-end fashion by minimizing the weighted combination of pixel-wise Euclidean loss and adversarial loss (given by (4.1)) between the estimated and ground truth density maps.

Inference details: Here, we describe the process to estimate the density map of a test image X^t_i. First, the global context map F^i_{tgc} for X^t_i is calculated in the following way. The test image X^t_i is divided into non-overlapping blocks of size $W^t_i/4 \times H^t_i/4$. All blocks are then fed into *GCE* to obtain their respective classification scores. As in training, the classification scores are used to build the context maps for each block to obtain the final global context feature map F^i_{tgc}. Next, the local context map F^i_{tlc} for X^t_i is calculated in the following way: A sliding window classifier (*LCE*) of size 64×64 is run across X^t_i and the classification scores from every window are used to build the local context F^i_{tlc}. Once the context information is obtained, X^t_i is fed into *DME* to obtain high-dimensional feature maps F^i_{tdme}. F^i_{tdme} is concatenated with F^i_{tgc} and F^i_{tlc} and fed into *F-CNN* to obtain the output density map. Note that due to additional context processing, inference using the proposed method is computationally expensive as compared to earlier methods such as [1, 104].

4.3 Experimental results

In this section, we present the experimental details and evaluation results on three publicly available datasets. First, the results of an ablation study conducted to demonstrate the effects of each module in the architecture is discussed. Along with the ablation study, we also perform a detailed comparison of the proposed method against a recent state-of-the-art-method [1]. This detailed analysis contains comparison of count metrics using MAE and MSE, along with qualitative and quantitative comparison of the estimated density maps. The quality of density maps is measured using two standard metrics: PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity in Image [146]). The ablation study is followed by a discussion and comparison of proposed method's results against several recent state-of-the-art methods on three datasets: ShanghaiTech [1], WorldExpo '10 [59] and UCF_CROWD_50 [6].

4.3.1 Ablation study using ShanghaiTech Part A

In this section, we perform an ablation study to demonstrate the effects of different modules in the proposed method. Each module is added sequentially to the network and results for each configuration are compared. Following four configurations are evaluated: (1) *DME*: The high-dimensional feature maps of *DME* are combined using 1×1 conv layer whose output is used to estimate the density map. L_E loss is minimized to train the network. (2) *DME* with only *GCE* and *F-CNN*: The output of *DME* is concatenated with the global context. *DME* and *F-CNN* are trained to estimate the density maps by

minimizing L_E loss. (3) *DME* with *GCE*, *LCE* and *F-CNN*. In addition to the third configuration, local context is also used in this case and the network is trained using L_E loss. (4) *DME* with *GCE*, *LCE* and *F-CNN* with $L_A + L_E$ (entire network). These results are compared with a fifth configuration: Zhang *et al.* [1] (which is a recent state-of-the-art method) in order to gain a perspective of the improvements achieved by the proposed method and its various modules. The evaluation is performed on Part A of ShanghaiTech [1] dataset.

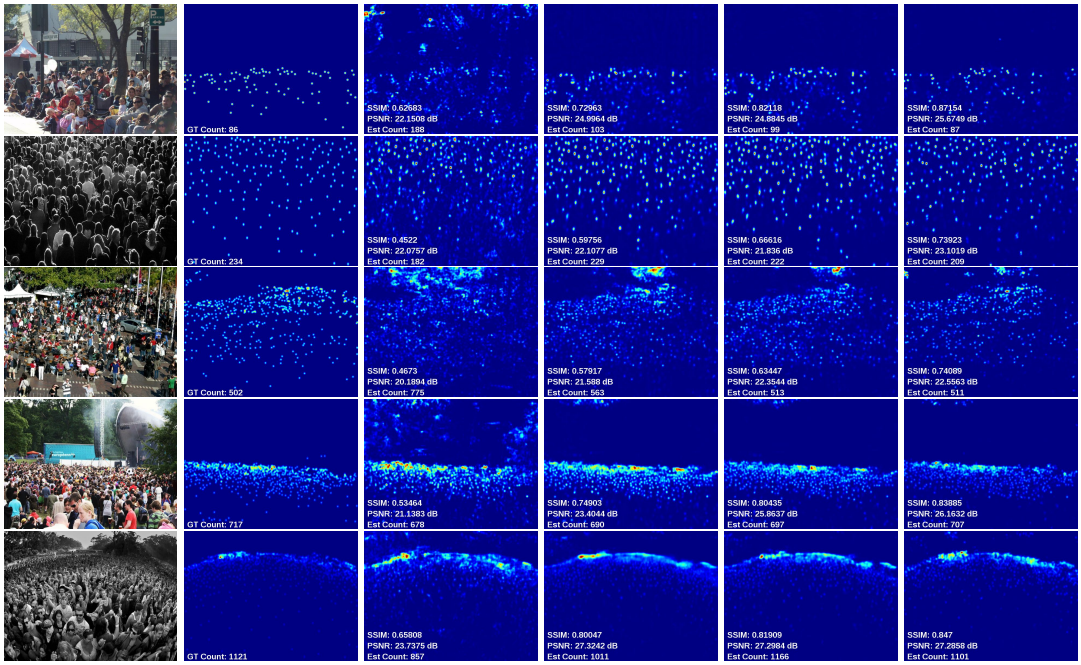


Figure 4.5: Comparison of results from different configurations of the proposed network along with Zhang *et al.* [1]. Top Row: Sample input images from the ShanghaiTech dataset. Second Row: Ground truth. Third Row: Zhang *et al.* [1]. (Loss of details can be observed). Fourth Row: *DME*. Fifth Row: *DME + GCE + F-CNN*. Sixth Row:*DME + GCE + LCE + F-CNN*. Bottom Row: *DME + GCE + LCE + F-CNN* with adversarial loss. Count estimates and the quality of density maps improve after inclusion of contextual information and adversarial loss.

Count estimation errors and quality metrics of the estimated density images for the various configurations are tabulated in Table 4.1. We make the

Table 4.1: Estimation errors for different configurations of the proposed network on ShanghaiTech Part A[1]. Addition of contextual information and the use of adversarial loss progressively improves the count error and the quality of density maps.

	Count estimation error		Density map quality	
Method	MAE	MSE	PSNR	SSIM
Zhang *et al.*[1]	110.2	173.2	20.91	0.52
DME	104.3	154.2	20.92	0.54
DME+GCE+FCNN	89.9	127.9	20.97	0.61
DME + GCE + LCE + FCNN	76.1	110.2	21.4	0.65
DME+GCE+LCE+ FCNN with L_A+L_E	**73.6**	**106.4**	**21.72**	**0.72**

following observations: (1) The network architecture for *DME* used in the proposed method is different from Zhang *et al.* [1] in terms of column depths, number of filters and filter sizes. These changes improve the count estimation error as compared to [1]. However, no significant improvements are observed in the quality of density maps. (2) The use of global context in (*DME + GCE + F-CNN*) greatly reduces the count error from the previous configurations. Also, the use of *F-CNN* (which is composed of fractionally-strided convolutional layers), results in considerable improvement in the quality of density maps. (3) The addition of local context and the use of adversarial loss progressively reduces the count error while achieving better quality in terms of PSNR and SSIM.

Estimated density maps from various configurations on sample input images are shown in Figure 4.5. It can be observed that the density maps generated using Zhang *et al.* [1] and *DME* (which regress on low-resolution maps) suffer from loss of details. The use of global context information and fractionally-strided convolutional layers results in better estimation quality. Additionally, the use of local context and minimization over a weighted

combination of L_A and L_E further improves the quality and reduces the estimation error.

4.3.2 Evaluations and comparisons

In this section, the results of the proposed method are compared against recent state-of-the-art methods on three challenging datasets.

ShanghaiTech. The proposed method is evaluated against four recent approaches: Zhang *et al.* [59], MCNN [1], Cascaded-MTL [103] and Switching-CNN [104] on Part A and Part B of the ShanghaiTech dataset are shown in Table 4.2. It can be observed from Table 4.2, that the proposed method is able to achieve superior results as compared to the other methods, which highlights the importance of contextual processing in our framework.

Table 4.2: Estimation errors on the ShanghaiTech dataset.

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Zhang *et al.* [59]	181.8	277.7	32.0	49.8
MCNN [1]	110.2	173.2	26.4	41.3
Cascaded-MTL [103]	101.3	152.4	20.0	31.1
Switching-CNN [104]	90.4	135.0	21.6	33.4
CP-CNN (ours)	**73.6**	**106.4**	**20.1**	**30.1**

WorldExpo'10. The proposed method is evaluated against five recent state-of-the-art approaches: Chen *et al.* [58], Zhang *et al.* [59], MCNN [1], Shang *et al.* [56] and Switching-CNN [104] is presented in Table 4.3. It can be observed from Table 4.3 that the proposed method outperforms existing approaches on an average while achieving comparable performance in individual scene estimations.

UCF_CC_50. Following the standard protocol discussed in [6], a 5-fold cross-validation was performed for evaluating the proposed method. Results are

Table 4.3: Average estimation errors on the WorldExpo'10 dataset.

Method	Scene1	Scene2	Scene3	Scene4	Scene5	Avgerage
Chen *et al.* [58]	**2.1**	55.9	**9.6**	11.3	**3.4**	16.5
Zhang *et al.* [59]	9.8	**14.1**	14.3	22.2	3.7	12.9
MCNN [1]	3.4	20.6	12.9	13.0	8.1	11.6
Shang *et al.* [56]	7.8	15.4	14.9	11.8	5.8	11.7
Switching-CNN [104]	4.4	15.7	10.0	11.0	5.9	9.4
CP-CNN (ours)	2.9	14.7	10.5	**10.4**	5.8	**8.86**

compared with seven recent approaches: Idrees *et al.* [6], Zhang *et al.* [59], MCNN [1], Onoro *et al.* [20], Walach *et al.* [17], Cascaded-MTL [103] and Switching-CNN [104]. It can be observed from Table 4.4 that our network achieves the lowest MAE and MSE count errors. This experiment clearly shows the significance of using context especially in images with widely varying densities.

Table 4.4: Estimation errors on the UCF_CC_50 dataset.

Method	MAE	MSE
Idrees *et al.* [6]	419.5	541.6
Zhang *et al.* [59]	467.0	498.5
MCNN [1]	377.6	509.1
Onoro *et al.* [20] Hydra-2s	333.7	425.2
Onoro *et al.* [20] Hydra-3s	465.7	371.8
Walach *et al.* [17]	364.4	341.4
Cascaded-MTL [103]	322.8.4	341.4
Switching-CNN [104]	318.1	439.2
CP-CNN (ours)	**295.8**	**320.9**

4.4 Summary

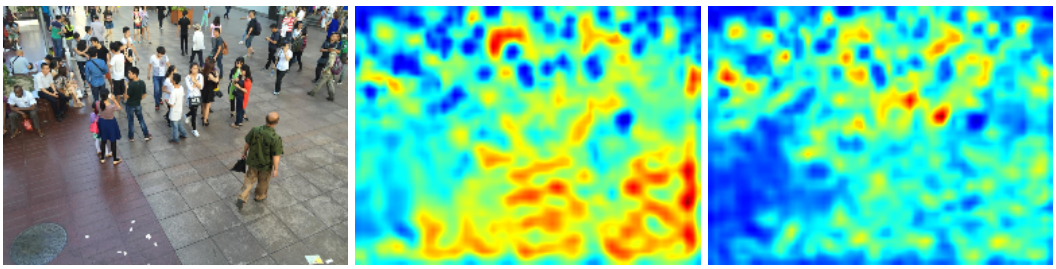
We presented contextual pyramid of CNNs for incorporating global and local contextual information in an image to generate high-quality crowd density maps and lower count estimation errors. The global and local contexts are obtained by learning to classify the input images and its patches into various

density levels. This context information is then fused with the output of a multi-column DME by a Fusion-CNN. In contrast to the existing methods, this work focuses on generating better quality density maps in addition to achieving lower count errors. In this attempt, the Fusion-CNN is constructed with fractionally-strided convolutional layers and it is trained along with the DME in an end-to-end fashion by optimizing a weighted combination of adversarial loss and pixel-wise Euclidean loss. Extensive experiments performed on challenging datasets and comparison with recent state-of-the-art approaches demonstrated the significant improvements achieved by the proposed method.

Chapter 5

Inverse Attention Guide Deep CNN for Crowd Counting

Crowd counting suffers from several challenges such as scale changes, heavy occlusion, illumination changes, clutter, non-uniform distribution of people, etc., making crowd counting and density estimation a very challenging problem, especially in highly congested scenes. Different techniques have been developed to address these issues. The issue of scale variations has received the most interest, with several works proposing different approaches such as multi-column networks [1], switching-cnns [104], use of context information [147], *etc*. While these methods provide significant improvements over recent techniques, the error rates of most of these methods are still far from optimal [20, 1]. A probable reason is that most of these methods train their networks from scratch and since the datasets have limited samples, they are unable to use high-capacity networks. For the few methods [147, 104] that achieve very low error rate, the training process is increasingly complex and requires multiple stages. For instance, Switching-CNN [104] involves different stages



(a) (b) (c)

Figure 5.1: Feature map visualization: (a) Input image, (b) Feature map before refinement, (c) Feature map after refinement using inverse attention. By infusing segmentation information via inverse attention into the counting network, we are able to suppress background regions, thus making the counting task much easier.

such as pre-training, differential training, switch training and coupled training. Similarly, CP-CNN [147] requires that their local and global estimators to be trained separately, followed by end-to-end training of their density map estimator. Although these methods achieve low error, their complex training process makes them hard to use.

Considering these drawbacks, our aim in the paper is to design a simple solution that is easy to train and achieves low count error. Given this objective, we start by presenting a VGG-16 based crowd counting network, which alone is able to achieve results that are comparable to recent state-of-the-art methods. While this baseline network achieves comparable performance with respect to recent methods, there is considerable room for further improvement. We present a simple, yet powerful technique that uses multi-task learning to further boost the counting performance. Specifically, we aim to efficiently infuse foreground/background segmentation mask into the counting network by simultaneously learning to count and segment. This use of related tasks for improving the performance is inspired by the success of recent multi-task

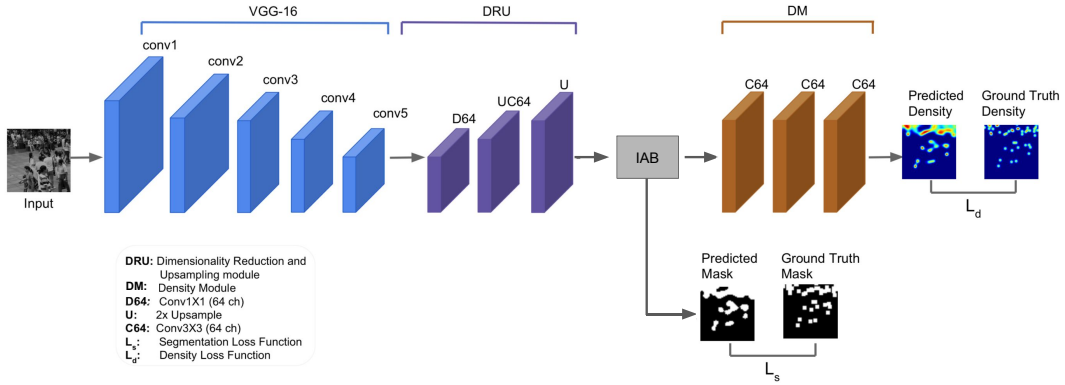


Figure 5.2: Overview of the proposed Inverse Attention Guided Deep Crowd Counting Network (IA-DCCN).

approaches such as Hyperface [134], instance aware semantic segmentation [133] and use of semantic segmentation for improving object detection.

Although this approach of infusing segmentation information through simple multi-task learning achieves considerable improvements in performance, it is limited by the fact that VGG16 is pre-trained on image net dataset and it will concentrate on regions with high response values during learning. To address this issue, we draw inspiration from the success of attention learning in various tasks such as action recognition, object recognition, image captioning, visual question answering *etc*[148, 149, 150, 151, 152, 153, 154, 155, 156], *etc*. Specifically, we propose an inverse attention module that captures important regions in the feature maps to focus on during learning. The inverted attention map enforces the network to focus specifically on relevant regions, thereby increasing the effectiveness of the learning mechanism.

5.1 Inverse Attention Guided Deep Crowd Counting Network (IA-DCCN)

Figure 7.2 provides an overview of the proposed Inverse Attention Guided Deep Crowd Counting Network (IA-DCCN) which is based on the VGG-16 network. We include an inverse attention block (IAB) with the objective of enriching the feature maps from VGG-16, thereby resulting in substantial improvements in the performance. This inverse attention block aims to encode segmentation masks into the feature maps due to which the counting task becomes considerably easier. Additionally, we employ a simple hard-mining technique, that effectively samples the training data due to which appreciable gains are observed. Details of the proposed method and its various components are described in the following sub-sections.

5.1.1 Base network

As illustrated in Figure 5.2, the base network consists of three parts: (i) first five convolutional blocks (conv1-conv5) from the VGG-16 architecture, (ii) dimensionality reduction and upsampling (DRU) module that reduces the dimensionality of feature maps from VGG-16 along the depth to 64 channels and upsamples them, and (iii) density module (DM) a set of three conv layers (with 64 channels and 3×3 filters) to perform the density estimation. Note that the network is fully convolutional and hence, it can be used on images of any size. The entire network regresses on the input image to produce a density map which indicates the number of people per pixel. This density map, when summed over all the pixels, provides an estimate of the number of people

in the input image. The conv layers belonging to VGG-16 architecture are initialized with pre-trained weights, where as the conv layers in the density estimation network are randomly initialized with $\mu = 0$ and $std = 0.01$. The network is trained by minimizing the following loss function:

$$L_d = \frac{1}{N} \sum_{i=1}^{N} \|F_d(X_i, \Theta) - D_i\|_2, \tag{5.1}$$

where, N is number of training samples, X_i is the i^{th} input image, $F_d(X_i, \Theta)$ is the estimated density, D_i is the i^{th} ground-truth density and it is calculated by summing a 2D Gaussian kernel centered at every person's location x_g as follows:

$$D_i(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma), \tag{5.2}$$

where σ is scale parameter of 2D Gaussian kernel and S is the set of all points at which people are located. Following [1], the density map generated by the network is $1/4^{th}$ of the input image resolution.

5.1.2 Segmentation infusion via Inverse Attention

The base-network, although very simple, achieves significantly low count errors and the results are better/comparable with respect to recent state-of-the-art methods [104, 147]. In order to further boost the performance, we propose to incorporate segmentation information into the counting network. A naive idea would be add to segmentation loss layer after an intermediate block in the network and train the network in a multi-task fashion. This would be similar to recent works like [134, 157, 157, 133] that learned different tasks simultaneously.

While this method results in a better performance as compared to the base network, we propose a more sophisticated method that uses inverse attention to incorporate segmentation information. For this, we draw inspiration from the recent work on tasks like image captioning, super-resolution, classification, visual question answering [148, 149] that use different forms of attention mechanisms to learn the features more effectively. Specifically, we introduce an inverse attention block (IAB) on top of the DRU module in the counting network as shown in Figure 5.2.

Fig 5.3 illustrates the mechanism of the inverse attention block. Specifically, the IAB takes feature maps (F) from the DRU as input and forwards them through a conv block CB_A to estimate background regions (which we call as inverse attention map - A^{-1}) in the input image. CB_A is defined by $\{\text{conv}_{512,32,1}\text{-relu-conv}_{32,32,3}\text{-relu-conv}_{32,1,3}\}$[1]. Feature maps F weighted by the inverse attention map are then subtracted from F to suppress the background regions *i.e.*,,

$$F' = F - F \odot A^{-1},$$

where F' is the attended feature map which is then forwarded through the density map module.

While the existing attention-based work learn the attention maps in a self-supervised manner, we instead use the ground-truth density maps to generate ground-truth inverse attention maps for supervising the inverse attention block. To generate the ground-truth, the density maps are thresholded and inverted. Note that by learning to estimate the background regions we are

[1] $conv_{N_i,N_o,k}$ denotes conv layer (with N_i input channels, N_o output channels, $k \times k$ filter size), *relu* denotes ReLU activation

automatically suppressing the background information from the feature maps of the DRU, hence, making it easier for the density module (DM) to learn the features more effectively. Figure 5.1 illustrates the feature maps before and after enrichment using IAB. It can be clearly observed that the use of inverse attention block aids in better feature learning.

The entire network is trained in a multi-task fashion by simultaneously minimizing the density loss and the segmentation loss. Formally, the overall loss (L) is defined as follows:

$$L = L_d + \lambda_s L_s, \tag{5.3}$$

where, L_d is the density loss (Equation 5.1), L_s is segmentation loss that is used for training the features of the inverse attention block, and λ_s is weighting factor for the segmentation loss. L_s is pixel-wise cross entropy error between estimated mask and ground-truth mask. Note that, the method does not require any additional labeling and it uses weakly annotated head/person regions based on the existing labels to compute the segmentation loss. The ground-truth mask is generated by thresholding the ground-truth density map. Bascially, the pixels that contain head regions are labeled as 1 (foreground), and otherwise as 0 (background). In spite of these annotations being noisy, the use of segmentation information results in considerable gains.

5.1.3 Hard sample mining (HSM)

Recent methods such as [158, 159] have demonstrated that effective sampling of data by selecting harder samples improves the classification performance

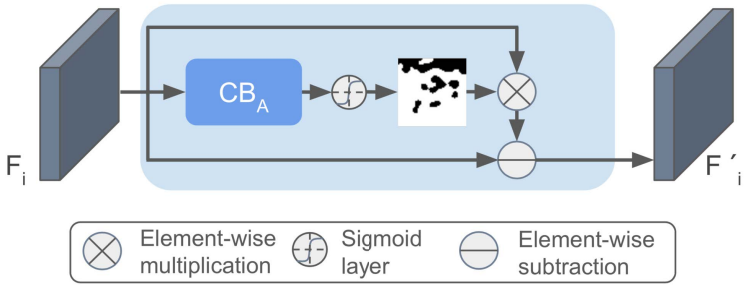


Figure 5.3: Inverse attention block.

of the network. Similar to these work, we employ an offline hard mining technique to train the network. This process, used to select samples from the training set every 5 epochs, involves the following steps: (i) compute the histogram of error on the entire training data, (ii) find the mode (T) of this error distribution (iii) training samples with $error > T$ are considered as hard samples and selected for training. This sample selection technique is effective in lowering the count error by an appreciable margin.

5.2 Experiments and results

In this section, we first describe the training and implementation specifics followed by a detailed ablation study to understand the effects of different components in the proposed network. We chose the ShanghaiTech dataset [1] to perform the ablation study as it contains significant variations in count and scale. Finally, we compare the results of the proposed method against several recent approaches on three publicly available datasets containing congested scenes. (ShanghaiTech, UCF_CROWD_50 [6], UCF-QNRF [2]).

5.2.1 Training and implementation details

The network is trained end-to-end using the Adam optimizer with a learning rate of 0.00005 and a momentum of 0.9 on a single NVIDIA GPU Titan Xp. 10 % of the training set is set aside for validation purpose. The final training dataset is formed by cropping patches of size 224×224 from 9 random locations from each image. Furthermore, data augmentation is performed by randomly flipping the images (horizontally) and adding random noise. Since the network is fully convolutional, image of any arbitrary size or resolution can be input to the network.

Table 5.1: Results of the ablation study on the ShanghaiTech Part A and Part B datasets. Figures in braces indicate the percentage improvement in error over previous configuration.

	Part A		Part B	
Configuration	MAE	MSE	MAE	MSE
Base network	76.7	119.1	17.3	22.5
Base network+S	71.2 (7.1%)	117.5 (1.3%)	15.0 (13.3%)	21.0 (6.7%)
Base network+IAB	68.1 (4.3%)	114.5 (2.5%)	13.6 (9.3%)	19.6 (6.7%)
Base network+IAB+HSM	66.9 (1.8%)	108.5 (5.2%)	10.2 (25.0%)	16.0 (18.3%)

5.2.2 Architecture ablation

To understand the effectiveness of the various modules present in the network, we perform experiments with the different settings using the ShanghaiTech dataset (Part A and Part B). This dataset consists of 2 parts with Part A containing 482 images and Part B containing 716 images and a total of 330,165 head annotations. Both parts have training and test subsets. Due to various challenges such as high density crowds, large variations in scales, presence of occlusion, etc, we chose to perform the ablation study on this dataset.

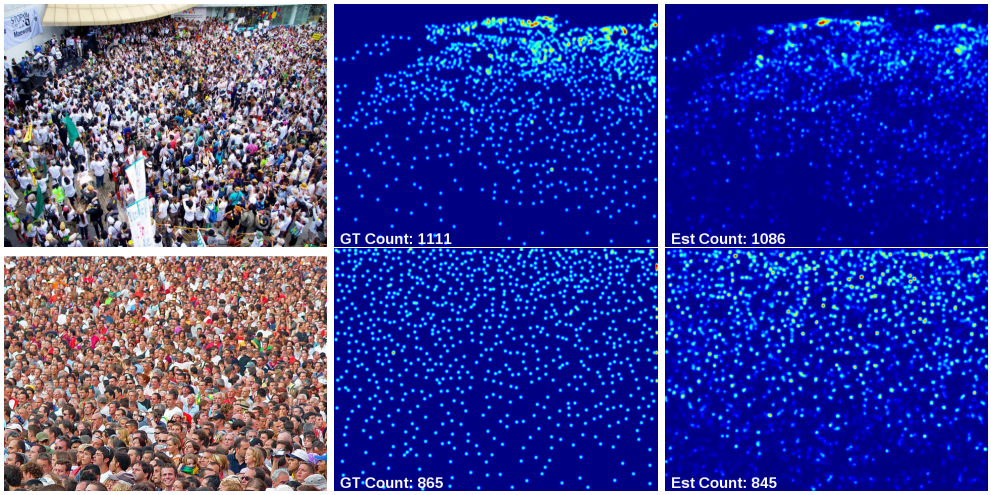
The results of these experiments are tabulated in Table 5.1. It can be observed that the base network, consisting of VGG-16 conv layers along with DRU module and density module (described in Section 5.1.1), does not provide the optimal performance. With the addition of the segmentation loss layer (Base network + S), we can observe an improvement of ∼7.1%/1.3% in MAE/MSE on Part-A and ∼15.0%/6.7% in MAE/MSE on Part-B over the base network. While this naive method of infusing segmentation network results in considerable improvements in the error, we show that there is still room for further improvements with the experiment where we incorporated the inverse attention block after the DRU module (Base network + IAB). The IAB module results in an improvement of ∼4.3%/2.5% in MAE/MSE on Part-A and ∼9.3%/6.7% in MAE/MSE on Part-B over the naive method.

Figure 5.1 visualizes the feature maps from the DRU module network with/without feature enrichment via segmentation infusion using inverse attention. Effects of incorporating segmentation information into the counting network can be clearly observed. This considerable reduction in the count error confirms our intuition that segmentation guided inverse can be used to aid the counting task, by introducing high-level foreground background knowledge into the feature maps of the VGG-16 network. λ_s in Equation 5.3 is set equal to 0.1 based on cross-validation.

Finally, we trained the entire network with hard sample mining (Base network + IAB + HSM as described in Section 5.1.3), where samples for training were selected based on the error at every fifth epoch. For this configuration, we observed an improvement of ∼1.8%/5.2% in MAE/MSE on Part A and

Table 5.2: Comparison of results on the ShanghaiTech dataset.

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Cascaded-MTL [103] (AVSS '17)	101.3	152.4	20.0	31.1
Switching-CNN [104] (CVPR '17)	90.4	135.0	21.6	33.4
TopDownFeedack [160] (AAAI '18)	97.5	145.1	20.7	32.8
CP-CNN [147] (ICCV '17)	73.6	**106.4**	20.1	30.1
IG-CNN [108] (CVPR '18)	72.5	118.2	13.6	21.1
CSR-Net [161] (CVPR '18)	68.1	115.0	10.6	16.0
IA-DCCN (ours)	**66.9**	108.4	**10.2**	**16.0**



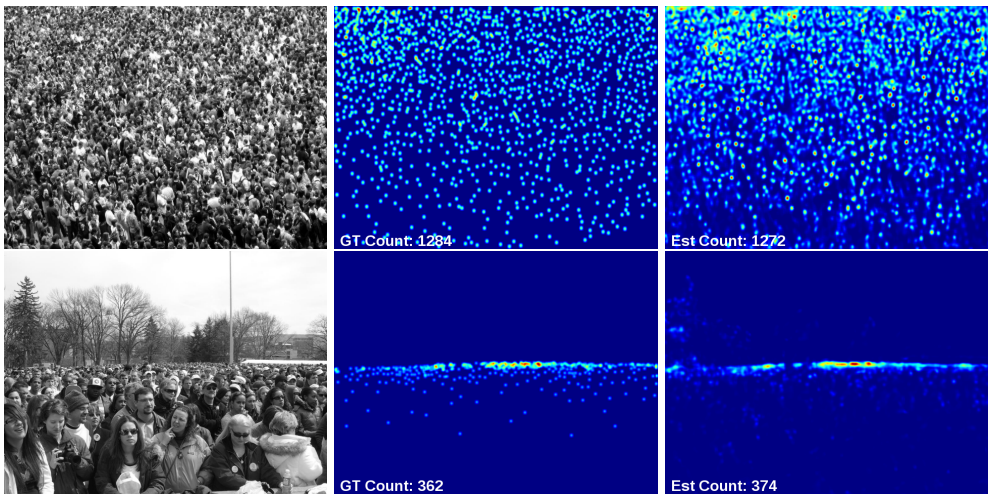
(a) (b) (c)

Figure 5.4: Sample results of the proposed method on the ShanghaiTech dataset [1]. *(a)* Input. *(b)* Ground truth *(c)* Estimated density map.

∼25.0%/18.3% in MAE/MSE on Part B over the base network with inverse attention. Hence, it can be concluded that hard sample mining is effective and provides appreciable gains on both parts of the dataset.

5.2.3 Comparison with recent methods

For comparison with various methods on different datasets, the entire network (IA-DCCN) is trained with hard sample mining.



(a) (b) (c)

Figure 5.5: Sample results of the proposed method on the UCF_CROWD_50 dataset [6]. *(a)* Input. *(b)* Ground truth *(c)* Estimated density map.

ShanghaiTech. The proposed method is compared with four recent approaches (Cascaded-MTL [103], Switching-CNN [104], CP-CNN [147], Top-down feedback [160], IG-CNN [108] and CSR-Net [161]) on Part A and Part B of the ShanghaiTech dataset and the results are presented in Table 5.2. The proposed IA-DCCN method achieves the lowest error rate in terms of MAE/MSE as compared to all recent methods on both parts of the dataset. Sample density estimation results are shown in Figure 5.4. From these results, it can be noted that the proposed method is able to achieve encouraging results while being simple to train as compared to existing approaches.

UCF_CROWD_50 . The UCF_CC_50 dataset [6] is a relatively smaller dataset with 50 annotated images of different resolutions and aspect ratios. We used the standard 5-fold cross-validation protocol discussed in [6] to evaluate the proposed method. Results are compared with several recent approaches:

Table 5.3: Comparison of results on the UCF_CROWD_50 dataset.

Method	MAE	MSE
Cascaded-MTL [103] (AVSS '17)	322.8	397.9
Switching-CNN [104] (CVPR '17)	318.1	439.2
CP-CNN [147] (ICCV '17)	295.8	**320.9**
TopDownFeedback [160] (AAAI '18)	354.7	425.3
IG-CNN [108] (CVPR '18)	291.4	349.4
CSR-Net [161] (CVPR '18)	266.1	397.5
IA-DCCN (ours)	**264.2**	394.4

Table 5.4: Comparison of results on the UCF-QNRF datastet.

Method	MAE	MSE
Idrees *et al.* [6] (CVPR '13)	315.0	508.0
Zhang *et al.* [59] (CVPR '15)	277.0	426.0
Cascaded-MTL [103] (AVSS '17)	252.0	514.0
Switching-CNN [104] (CVPR '17)	228.0	445.0
Idrees *et al.* [2] (ECCV '18)	132.0	191.0
IA-DCCN (ours)	**125.3**	**185.7**

Cascaded-MTL [103], Switching-CNN [104], CP-CNN [147], Top-down feedback [160], IG-CNN [108] and CSR-Net [161]. The results are tabulated in Table 5.3. It can be observed that the proposed method achieves the lowest MAE error as compared to the recent methods. Although the proposed approach performs slightly worse in terms of MSE as compared to CP-CNN [147], it is important to note that the MSE error is comparable to the existing approaches. Additionally, we believe that these results are especially significant considering the simplicity of the proposed approach. Sample density estimation results are shown in Figure 5.5.

UCF-QNRF: The UCF-QNRF [2] is a recent dataset that contains around 1200 images with approximately 1.2 million annotations. The results of the proposed method on this dataset as compared with recent methods ([6],[1],[103])

are shown in Table 5.4. The proposed method is compared against five different approaches: [6], [1], [103],[104], and [2]. It can be observed that the proposed method outperforms other methods by a considerable margin.

5.2.3.1 Inference speed

To evaluate the inference speed of the proposed approach, we run IA-DCCN on our machine which is equipped with Intel Xeon E5-2620v4@2.10GHz and an NVIDIA Titan Xp GPU. The run times are reported in Table 5.5 for different resolutions ranging from 320×240 to 1600×1200. It can be observed that the proposed method is efficient and is able to run at ∼76 fps while processing high resolution images (1600×1200). Note that the majority of processing time is taken up by the VGG-16 network.

Table 5.5: Inference time for different resolutions in msec.

Res (W×H)	320×240	640×480	1280×960	1600×1200
IA-DCCN (ours)	2.7	4.8	8.9	13.1

5.3 Summary

We presented a very simple, yet effective crowd counting approach based on the VGG-16 network and inverse attention, referred to as Inverse Attention Guided Deep Crowd Counting Network (IA-DCCN). The proposed approach aims to infuse segmentation information into the counting network via an inverse attention mechanism. This infusion of segmentation maps into the network enriches the feature maps of VGG-16 network due to which the background information in the feature maps get suppressed, making the counting task rather easier. In contrast to existing approaches that employ

complex training process, the proposed approach is a single-stage training framework and achieves significant improvements over the recent methods while being computationally fast.

Chapter 6

HA-CCN: Hiearcichal Attention Based Crowd Counting Network

We propose to improve the counting performance by explicitly modeling spatial pixel-wise attention and global attention into the counting network. Considering that crowd images have large variations in head sizes, it is essential to leverage multi-scale information by employing feature maps from different conv layers of the VGG16 network [140]. Several works such as [162, 163, 164, 165] have demonstrated that different sized objects are captured by different layers in a deep network. Hence, an obvious approach would be to design a multi-scale counting network [166, 167] that concatenates feature maps from different layers of the VGG16 network. However, earlier layers in a deep network capture primitive features and do not learn semantic awareness. Due to this, naive concatenation of feature maps from different layers of the network is not necessarily an optimal approach to address the issue of large scale variations in crowd images.

To address this issue, we introduce a spatial attention module (SAM) in the network, that is designed to infuse semantic awareness into the feature

maps. This module takes the feature maps from lower layers as input, and learns to perform foreground-background segmentation. Furthermore, it uses this learned segmentation map to enhance the lower layer feature maps by selectively attending to specific spatial locations in this lower layer. Furthermore, we also attempt to augment channel-wise information in the higher level layers by employing a set of global attention modules. These modules selectively enhance important channels while suppressing the unnecessary ones.

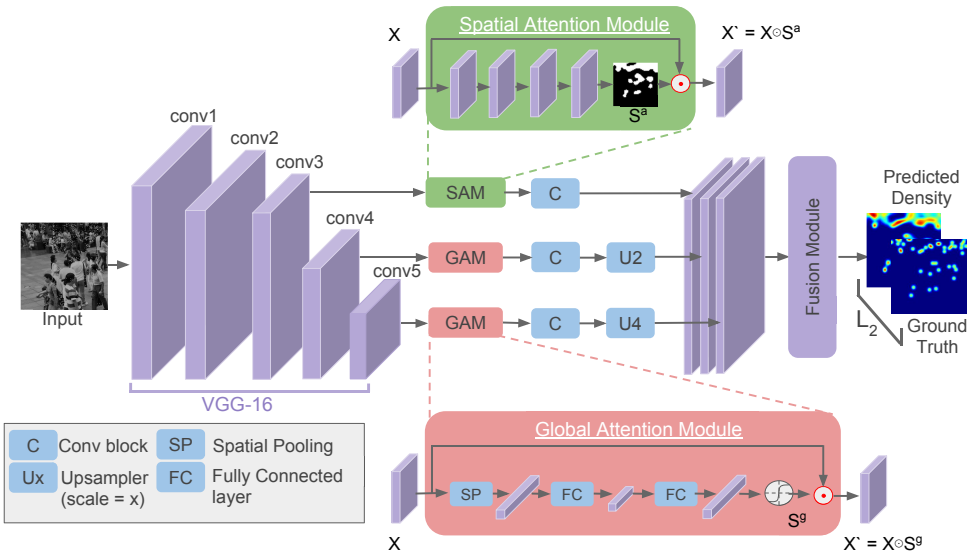


Figure 6.1: Overview of the proposed Hierarchical Attention-based Crowd Counting Network (HA-CCN). VGG16 is used as the base network. Feature maps from conv3 are forwarded through a spatial attention module that incorporates pixel-wise segmentation information into the features. Feature maps form higher layers (conv4, conv5) are forwarded through a set of global attention modules that augment the feature maps along the channel dimension.

6.1 Hierarchical attention for crowd counting

A natural solution to address scale variation in crowd counting images is to leverage multi-scale features from different layers in the backbone network. However, the layers in the backbone network are learned in a hierarchical manner with the earlier layers capturing primitive features and the subsequent layers capturing higher level concepts. Hence, direct fusion of these multi-scale feature maps might not be the most effective approach. In order to overcome this, we propose Hierarchical Attention-based Crowd Counting Network (HA-CCN) that leverages attention mechanisms to enrich features from different layers of the network for more effective multi-scale fusion.

Figure 6.1 provides an overview of the proposed method, which is based on the VGG-16 network. We include a spatial attention module (SAM) and a set of global attention modules (GAM) with the objective of enriching the feature maps at different levels. The base network consists of conv layers (conv1 $\sim$ conv5) from the VGG16 network. The conv3 features are enhanced by passing them through SAM. Similarly, features from conv4 and conv5 are passed through GAMs in order to perform channel-wise enhancement. The enhanced feature maps from conv3 are then forwarded through a conv block which consists of 3 conv layers defined as follows: {Conv2d(256,64,1)^2-ReLU, Conv2d(64,64,3)^2-ReLU, Conv2d(64,24,1)^2-ReLU}.

Similarly, the enhanced features from conv4 and conv5 are forwarded through a conv block and an upsampling layer to scale the feature maps to a size similar to that of conv3 feature maps. The conv block is defined by: {Conv2d(512,64,1)^2-ReLU, Conv2d(64,64,3)^2-ReLU, Conv2d(64,24,1)^2-ReLU}.

These processed features are concatenated together before being forwarded through the fusion module that consists of a set of conv layers to produce the final density map. These conv layers are defined by: {Conv2d(72,64,1)[2]-ReLU, Conv2d(64,64,3)[2]-ReLU, Conv2d(64,1,1)[2]-ReLU}. The network is trained by minimizing the Euclidean distance between the predicted density map and the ground truth density map as below:

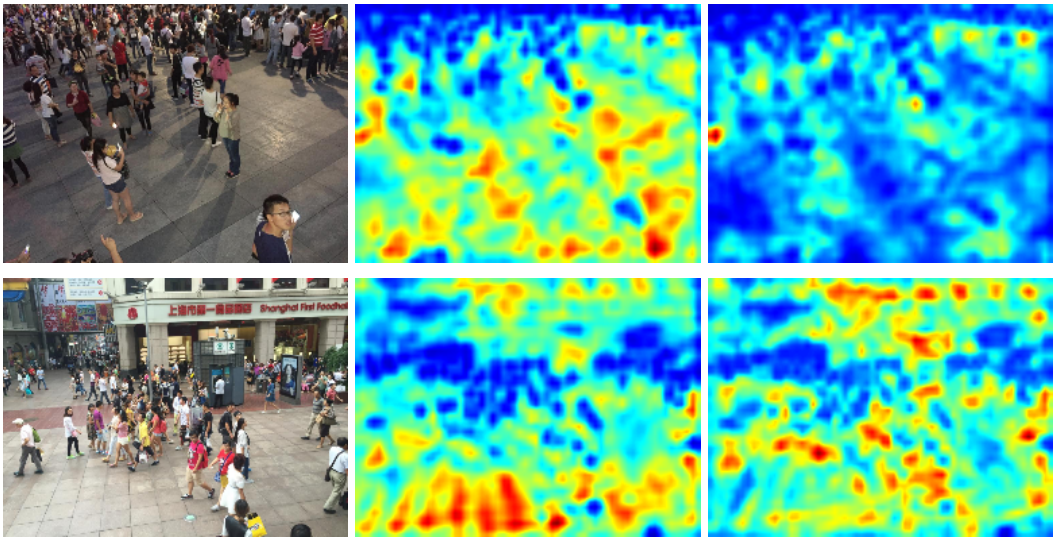
$$L_d = \frac{1}{N}\sum_{i=1}^{N} \|F_d(X_i, \Theta) - D_i\|_2, \tag{6.1}$$

where, N is number of training samples, X_i is the i^{th} input image, $F_d(X_i, \Theta)$ is the estimated density, D_i is the i^{th} ground-truth density and it is calculated by summing a 2D Gaussian kernel centered at every person's location x_g as follows: $D_i(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma)$, where σ is scale parameter of 2D Gaussian kernel and S is the set of all points where people are located. The density map generated by the network is $1/4^{th}$ of the input image resolution. Due to its construction, the sum of the density map provides an estimate of the number of people in the input image.

Details of the proposed method and its various components are described in the following sub-sections.

6.1.1 Spatial attention module

Inspired by the success of spatial attention mechanisms in image captioning, visual question answering and classification [168, 169, 148], we explore its utilization for crowd counting. The goal of spatial attention is to select attentive regions in the feature maps, which are then used to dynamically enhance the



(a) (b) (c)

Figure 6.2: Visualization of the conv3 feature maps: (a) Input image (b) Before segmentation infusion (c) After segmentation infusion. By infusing segmentation information into the counting network, we are able to suppress background regions. Note that in the density maps, red color indicates high density and blue color indicates low density.

feature responses. In contrast to existing work that learn spatial attention in a self-supervised manner, we propose to learn this by explicitly using foreground background segmentation for supervision. Since the goal of the spatial attention module is to focus on relevant regions and foreground regions are necessarily a part of these relevant regions, it is beneficial to use these labels to supervise the module. By explicitly supervising the module, we are able to infuse foreground background information into the network, thereby forcing the network to focus on relevant regions among the foreground. Moreover, these labels are readily available and hence, it does not require additional annotation efforts.

The spatial attention module consists of 4 conv layers with 3×3 filters that

takes feature maps from the conv3 layer of the VGG16 network as input (denoted by $X \in R^{W \times H \times C}$), and produces a segmentation output $S^a \in [0,1]^{W \times H}$. The segmentation map is then used to actuate the low level feature map X via element-wise multiplication: $\hat{X} = X \odot S^a$, where $\hat{X}$ is the actuated low level feature map from conv3. Through this attention mechanism, we are able to incorporate segmentation awareness into the low level feature maps. As illustrated in Figure 6.2, the use of segmentation information into the network enriches feature maps by suppressing irrelevant regions and boosting the foreground regions. The actuated feature maps are then forwarded to the fusion block (FM), where they are fused with the features from other layers to generate the final density map.

The weights of SAM are learned by minimizing the cross entropy error between the predicted segmentation map and the corresponding ground-truth. Normally, the segmentation task requires pixel-wise annotations. However, in this case existing ground truth density map annotations are thresholded to generate the ground truth segmentation maps, which are then used to train the spatial attention module. Basically, the pixels that contain head regions are labeled as 1 (foreground), and otherwise as 0 (background). Hence, the proposed method does not require any additional labeling. In spite of these annotations being noisy, the use of segmentation information results in considerable gains.

6.1.2 Global attention modules

In contrast to the spatial attention module that attends to relevant spatial locations in the feature maps of low-level layers, the global attention module (GAM) is designed to attend to feature maps in the channel-dimension. The global attention module is similar to the channel-wise attention used in earlier work like [148, 170]. Specifically, this module consumes feature maps from the backbone network and learns to compute attention along the channel dimension. The computed attention captures the important channels in the feature maps and hence aids in suppressing information from unnecessary channels. Since this module operates at a global level in terms of spatial dimension, we refer to this attention module as global attention module. It has been demonstrated in [148, 168, 171], that channels capture the presence either different parts of an object or different classes of objects and channel-wise attention is an effective way to boost the correlation between object/object parts and image captions.

Based on these considerations, we employ a set of global attention modules, which take feature maps from the higher conv layers as input and produce a channel-wise attention map, which is then used to actuate the feature maps along the channel dimension. Mathematically, given a feature map input $X \in R^{W \times H \times C}$, GAM first performs a spatial pooling to produce pooled features $Y \in R^{1 \times 1 \times C}$ using

$$Y_i = \frac{1}{W \times H} \sum_{w,h} X_i^{wh}, \tag{6.2}$$

where i is the channel index, and w, h are spatial indices. Y is passed

through a set of fully-connected (FC) layers defined by $FC(512, 64) - ReLU - FC(64, 64) - ReLU - FC(64, 512)$[1] and a sigmoid layer to produce channel-wise attention vector $S^g \in R^{1\times1\times C}$. Finally, S^g is used to actuate the feature maps from the higher layer by performing a element-wise multiplication, *i.e.*,$\hat{X} = X \odot S^g$.

6.2 Experiments and results

In this section, we first describe the training and implementation specifics followed by a detailed ablation study to understand the effects of different components in the proposed counting network. Finally, we compare results of the proposed method against several recent approaches on 3 publicly available datasets (ShanghaiTech [1], UCF-QNRF [2], UCF_CROWD_50 [6]).

6.2.1 Training and implementation details

The network is trained end-to-end using the Adam optimizer with a learning rate of 0.00005 and a momentum of 0.9 on a single NVIDIA GPU Titan Xp. 10 % of the training set is set aside for validation purpose. The final training dataset is formed by cropping patches of size 224×224 from 9 random locations from each image. Further data augmentation is performed by randomly flipping the images (horizontally) and adding random noise.

Since the network is fully convolutional, entire test image is forwarded

[1] FC_{N_i,N_o} denotes fully connected layer (with N_i input elements, N_o output elements)

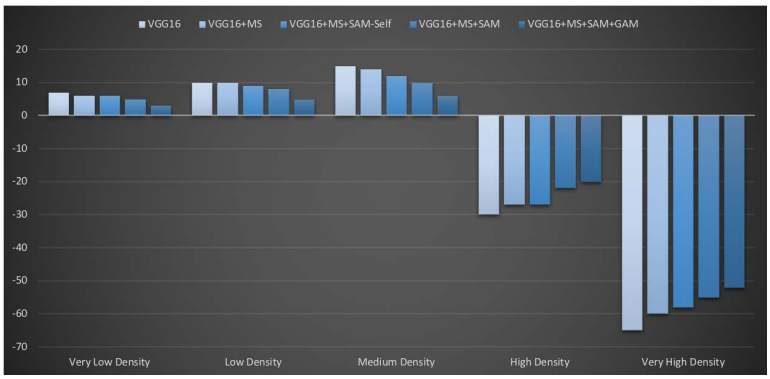


Figure 6.3: Ablation study: MAE for different configurations at different density levels.

Table 6.1: Results of the ablation study on ShanghaiTech Part A and Part B datasets.

	Part A		Part B	
Configuration	MAE	MSE	MAE	MSE
VGG16	78.3	120.1	18.3	22.9
VGG16+MS	72.1	115.5	15.6	20.6
VGG16+MS+SAM (Self-sup)	69.5	108.2	12.3	20.1
VGG16+MS+SAM	65.1	103.5	10.6	19.6
VGG16+MS+SAM+GAM (HA-CCN)	**62.9**	**94.9**	**8.1**	**13.4**

through the network during inference. This results in faster inference as compared to the existing methods (such as Switching-CNN [104], IG-CNN [108], CP-CNN [147], SA-Net [107]) which involve patch-based testing.

6.2.2 Architecture ablation

To understand the effectiveness of various modules present in the network, we perform experiments with the different settings using ShanghaiTech dataset (Part A and Part B). This dataset consists of 2 parts with Part A containing 482 images and Part B containing 716 images and a total of 330,165 head annotations. Both parts have training and test subsets.

The ablation study consisted of evaluating 3 baselines in addition to the proposed method:

(i)*VGG16*: VGG16 network with an additional conv block at the end.

(ii)*VGG16+MS*: VGG16 with multi-scale feature map concatenation and a fusion module at the end.

(iIi)*VGG16+MS+SAM (Self-sup)*: VGG16 with spatial attention module (self-supervised) for conv3 layer and multi-scale feature map concatenation, followed by a fusion module at the end.

(iv)*VGG16+MS+SAM*: VGG16 with spatial attention module for conv3 layer and multi-scale feature map concatenation, followed by a fusion module at the end.

(v)*VGG16+MS+SAM+GAM (HA-CCN)*: proposed method.

The results of these experiments are tabulated in Table 6.1. It can be observed that the naive approach of performing multi-scale feature concatenation does not necessarily yield the most optimal performance. The use of SAM infuses segmentation information in to the feature maps of conv3 layer in the base network, resulting in considerable reduction of the count error as compared to the naive approach. The use of global attention results in further improvement, thus showing significance of incorporating channel-wise importance in the network.

Additionally, it can also be noted that the explicitly supervised SAM results in better performance as compared to the self-supervised spatial attention.

Table 6.2: Comparison of results on the ShanghaiTech [1] and UCF_CROWD_50 [3] datasets. Top two methods are highlighted using underline and bold fonts respectively. * indicates patch-based testing.

	ShTech-A		ShTech-B		UCF-CROWD	
Method	MAE	MSE	MAE	MSE	MAE	MSE
Switch-CNN [104]* (CVPR '17)	90.4	135.0	21.6	33.4	318.1	439.2
CP-CNN [147]* (ICCV '17)	73.6	106.4	20.1	30.1	295.8	**320.9**
IG-CNN [108]* (CVPR '18)	72.5	118.2	13.6	21.1	291.4	349.4
ACSCP [109] (CVPR '18)	75.7	102.7	17.2	27.4	291.0	404.6
CSRNet [161] (CVPR '18)	68.2	115.0	10.6	16.0	266.1	397.5
ic-CNN [112] (ECCV '18)	69.8	117.3	10.7	16.0	260.9	365.5
SA-Net [107]* (ECCV '18)	67.0	104.5	8.4	13.6	**258.5**	**334.9**
IA-DCCN [116]* (AVSS '19)	66.9	108.4	10.2	16.0	264.2	394.4
ADCrowdNet [113] (CVPR '19)	63.2	98.9	**8.2**	15.7	266.4	358.0
RReg [114] (CVPR '19)	**63.1**	**96.2**	8.7	**13.5**	-	-
HA-CCN (ours)	**62.9**	**94.9**	**8.1**	**13.4**	**256.2**	348.4

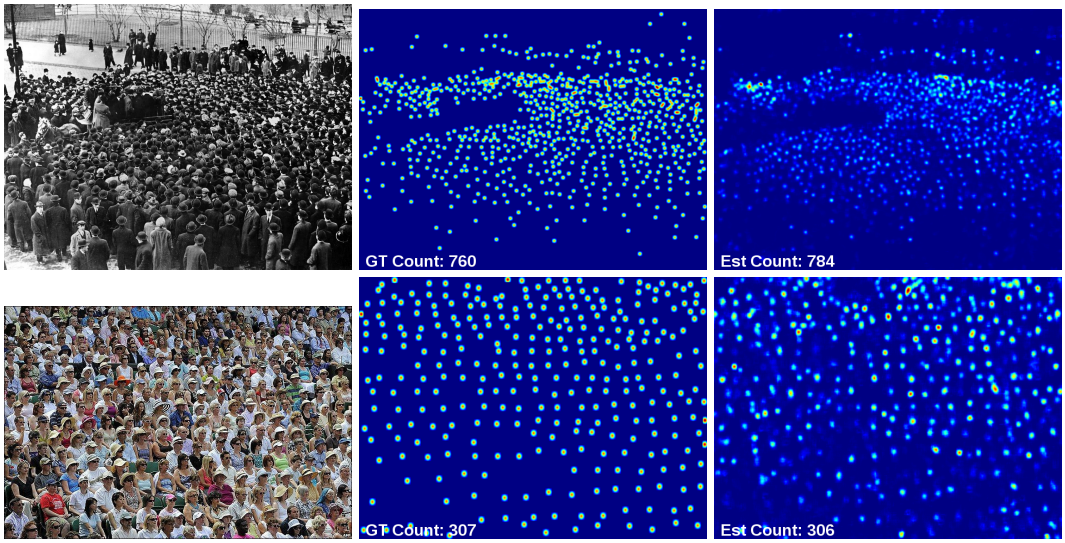
Table 6.3: Comparison of results on the UCF-QNRF datastet [2]. Top two methods are highlighted using underline and bold fonts respectively.

Method	MAE	MSE
Idrees *et al.* [6] (CVPR '13)	315.0	508.0
Zhang *et al.* [59] (CVPR '15)	277.0	426.0
CMTL *et al.* [103] (AVSS '17)	252.0	514.0
Switching-CNN [104] (CVPR '17)	228.0	445.0
Idrees *et al.* [2] (ECCV '18)	132.0	191.0
IA-DCCN *et al.* [116] (AVSS '19)	**125.3**	**185.7**
HA-CCN (ours)	**118.1**	**180.4**

Figure 6.3 shows a plot of the mean absolute error for different configurations in the ablation study at different density levels. It can be observed that the proposed HA-CCN network achieves best error among all the density levels.

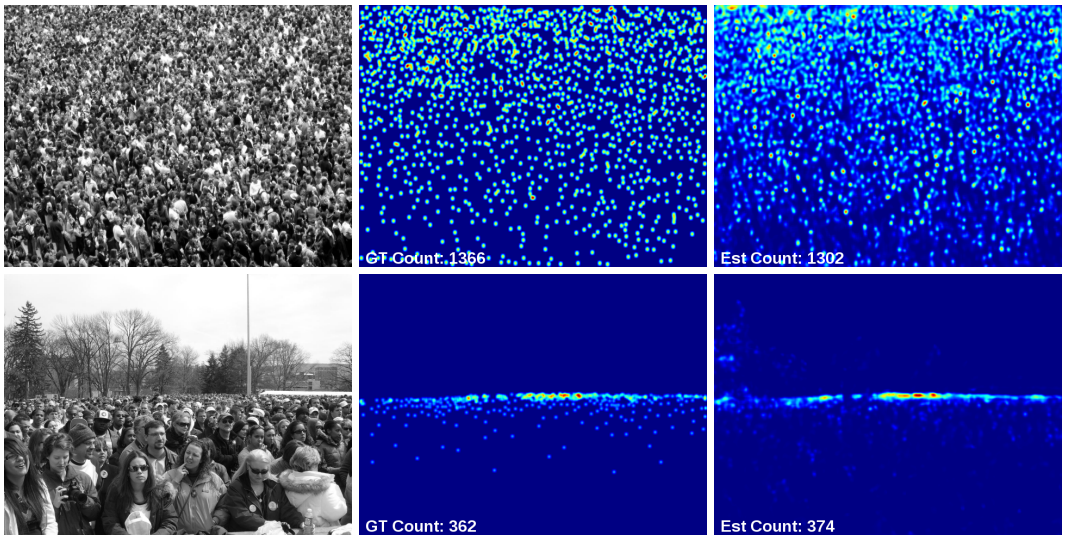
6.2.3 Comparison with recent methods

In this section, we discuss the results of the proposed method as compared with recent approaches on 3 different datasets: ShanghaiTech [1], UCF_CROWD_50 [6] and UCF-QNRF [2]. As discussed earlier, ShanghaiTech has 2 parts with



(a) (b) (c)

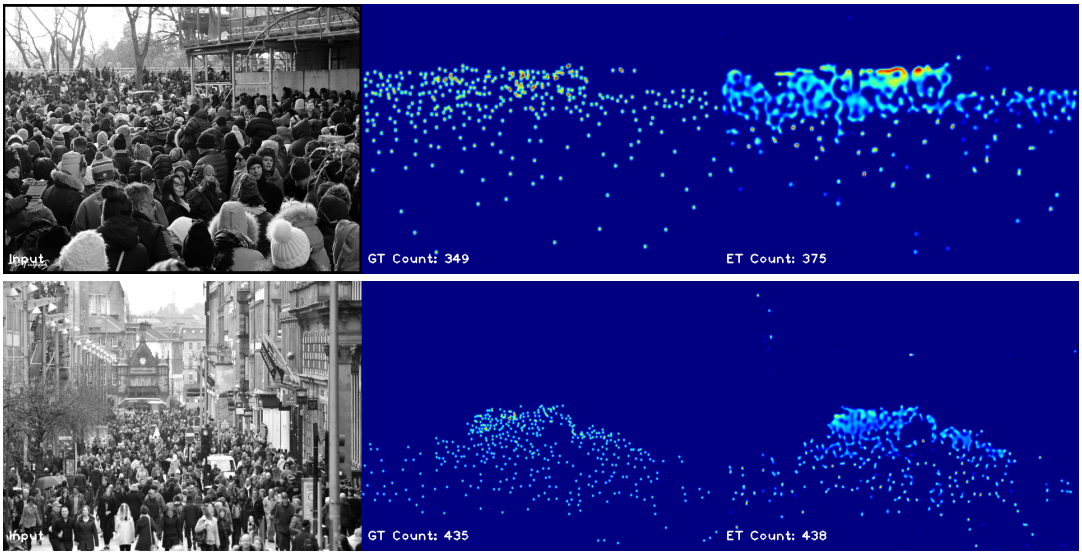
Figure 6.4: Sample results of the proposed method on ShanghaiTech [1] *(a)* Input. *(b)* Ground truth *(c)* Estimated density map.



(a) (b) (c)

Figure 6.5: Sample results of the proposed method on UCF_CROWD_50 [6]. *(a)* Input. *(b)* Ground truth *(c)* Estimated density map.

a total of 1198 images. The UCF_CC_50 dataset [6] contains 50 annotated images of different resolutions and aspect ratios. Following the standard



(a) (b) (c)

Figure 6.6: Sample results of the proposed method on UCF-QNRF dataset [2]. *(a)* Input. *(b)* Ground truth *(c):* Estimated density map.

protocol discussed in [6], a 5-fold cross-validation is performed for evaluating the proposed method. UCF-QNRF [2] is a more recent dataset that contains 1,535 high quality images with a total of 1.25 million annotations. The training and test sets consists of 1201 and 334 images respectively.

Table 6.2 shows the results of the proposed method on the ShanghaiTech and UCF_CROWD_50 datasets as compared with several recent approaches: Switching-CNN [104], CP-CNN [147], IG-CNN [108], D-ConvNet [110], Liu *et al.* [111], CSRNet [161], ic-CNN [112], SA-Net [107], ADCrowdNet [113] and Residual Regression [114]. It can be observed that the proposed method outperforms all existing methods.

Table 6.3 shows the comparison of results on the recently released large-scale UCF-QNRF [2] dataset. The proposed method is compared against five different approaches: Idrees *et al.* [6], MCNN [1], CMTL [103], Switching-CNN

[104] and Idrees *et al.* [2]. It can be observed that the proposed method is able to achieve state-of-the-art results on this complex dataset. Figure 6.4, 6.5 and 6.6 illustrate the qualitative results for sample images from the ShanghaiTech, UCF_CROWD_50 and UCF-QNFRF datasets respectively.

6.2.4 Cross dataset performance

We compare the generalization abilities of the proposed method with that of recent methods (MCNN [1], Switching-CNN [104], D-ConvNet [110]) by testing the network (trained on ShanghaiTech A dataset) on target datasets such as ShanghaiTech B, UCF_CROWD_50 and WorldExpo '10 [59]. The results are presented in Table 6.4. Note that the other networks are also trained on ShanghaiTech A dataset. The cross-dataset performance is measured using the overall count error (MAE/MSE) and the drop in performance. The drop in performance is the difference between the error of the model trained on the target set and that of the model trained on source set, when tested on target set. It can be observed that the proposed method is relatively more robust to change in dataset distribution as compared to the other methods.

Although the proposed method demonstrates better cross-dataset performance as compared to existing methods, there is considerable gap in the performance as compared to when the network is fully supervised on the target set.

Table 6.4: Cross dataset performance. S: Model is trained on target set, NS: Model is trained on source and tested on target set. C: Drop in performance between S and NS.

Method	Target Set				
	ShanghaiTech B		UCF_CROWD_50		WExpo '10
	MAE (S/NS/C)	MSE (S/NS/C)	MAE (S/NS/C)	MSE (S/NS/C)	MAE (S/NS/C)
MCNN [1]	26.4/39.6/13.2	41.3/102.5/61.2	377.6/397.7/20.1	509.1/624.1/115.0	11.6/25.2/13.6
Switch CNN [104]	21.6/59.4/37.8	33.4/130.7/97.3	318.1/1117.5/799.4	439.2/1315.4/876.2	9.4/31.1/21.7
D-ConvNet [110]	18.7/49.1/30.4	26.0/99.2/73.2	288.4/364.0/75.6	404.7/545.8/141.1	-
HA-CCN (ours)	8.1/29.1/21.0	13.4/74.1/60.1	256.2/339.8/83.6	348.4/463.2/114.8	8.5/22.0/13.5

6.3 Summary

In this work, we presented a crowd counting network that consists of different attention mechanisms at various levels in the network. Specifically, the proposed network involves two sets of attention modules: spatial attention and global attention module. The spatial attention module incorporates pixel level attention through a way of foreground background segmentation into the features of the earlier layers of the network. The global attention module incorporates channel-wise importance into the network.

Chapter 7

Multi-Level Bottom-Top and Top-Bottom Feature Fusion for Crowd Counting

Crowd counting from a single image, especially in congested scenes, is a difficult problem since it suffers from multiple issues like high variability in scales, occlusions, perspective changes, background clutter, etc. Recently, several convolutional neural network (CNN) based methods [59, 1, 104, 147, 111, 110, 109, 108, 107, 112] have attempted to address these issues with varying degree of successes. Among these issues, the problem of scale variation has particularly received considerable attention from the research community. Scale variation typically refers to large variations in scale of the objects being counted (in this case heads) (i) within image and (ii) across images in a dataset. Several other related tasks like object detection [163, 172, 173, 162, 9, 174] and visual saliency detection [175, 176, 177, 156] are also affected by such effects. However, these effects are more evident especially in crowd counting in congested scenes. Furthermore, since the annotation process for highly congested

scenes is notoriously challenging, the datasets available for crowd counting typically provide only x, y location information about the heads in the images. Since the scale labels are unavailable, training the networks to be robust to scale variations is much more challenging. Here, we focus on addressing the issue of scale variation and missing scale information from the annotations.

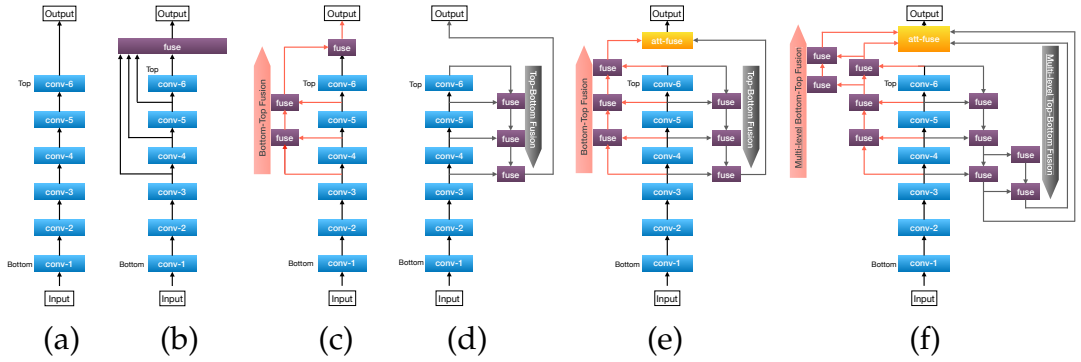


Figure 7.1: Illustration of different multi-scale fusion architectures: (a) No fusion, (b) Fusion through concat or add, (c) Bottom-top fusion, (d) Top-bottom fusion, (e) Bottom-top and top-bottom fusion, (f) Multi-level bottom-top and top-bottom fusion (proposed).

CNNs are known to be relatively less robust to the presence of such scale variations and hence, special techniques are required to mitigate their effects. Using features from different layers of a deep network is one approach that has been successful in addressing this issue for other problems like object detection. It is well known that feature maps from shallower layers encode low-level details and spatial information [163, 178, 179, 180, 181], which can be exploited to achieve better localization. However, such features are typically noisy and require further processing. Meanwhile, deeper layers encode high-level context and semantic information [163, 178, 179, 180] due to their larger receptive field sizes, and can aid in incorporating global context into the

network. However, these features lack spatial resolution, resulting in poor localization. Motivated by these observations, we believe that high-level global semantic information and spatial localization play an important role in generating effective features for crowd counting, and hence, it is important to fuse features from different layers in order to achieve lower count errors.

In order to perform an effective fusion of information from different layers of the network, we explore different fusion architectures as shown in Figure 7.1*(a)-(d)*, and finally arrive at our proposed method (Figure 7.1*(f)*). Figure 7.1*(a)* is a typical deep network which processes the input image in a feed-forward fashion, with no explicit fusion of multi-scale features. The network in Figure 7.1*(b)* extracts features from multiple layers and fuses them simultaneously using a standard approach like addition or concatenation. With this configuration, the network needs to learn the importances of features from different layers automatically, resulting in a sub-optimal fusion approach. As will be seen later in Section 7.3.2, this method does not produce significant improvements as compared to the base network.

To overcome this issue, one can choose to progressively incorporate detailed spatial information into the deeper layers by sequentially fusing the features from lower to higher layers (bottom-top) as shown in Figure 7.1*(c)* [117]. This fusion approach explicitly incorporates spatial context from lower layers into the high-level features of the deeper layers. Alternatively, a top-bottom fusion (Figure 7.1*(d)*) [160] may be used that involves suppressing noise in lower layers, by propagating high-level semantic context from deeper layers into them. These approaches achieve lower counting errors as compared to the

earlier configurations. However, both of these methods follow uni-directional fusion which may not necessarily result in optimal performance. For instance, in the case of bottom-top fusion, noisy features also get propagated to the top layers in addition to spatial context. Similarly, in the case of top-bottom fusion, the features from the top layer may end up suppressing more than necessary details in the lower layers. Variants of these top-bottom approaches and bottom-top approaches have been proposed for other problems like semantic segmentation and object detection [182, 183, 184, 185].

Recently, a few methods [186, 187] have demonstrated superior performance on other tasks by using multi-directional fusion technique (Figure 7.1*(e)*) as compared to uni-directional fusion. Motivated by the success of these methods on their respective tasks, we propose a multi-level bottom-top and top-bottom fusion (MBTTBF) technique as shown in Fig 7.1*(f)*. By doing this, more powerful features can be learned by enabling high-level context and spatial information to be exchanged between scales in a bidirectional manner. The bottom-top path ensures flow of spatial details into the top layer, while the top-bottom path propagates context information back into the lower layers. The feedback through both the paths ensures that minimal noise is propagated to the top layer in the bottom-top direction, and also that the context information does not over-suppress the details in the lower layers. Hence, we are able to effectively aggregate the advantages of different layers and suppress their disadvantages. Note that, as compared to existing multi-directional fusion approaches [186, 187], we propose a more powerful

fusion technique that is multi-level and aided by scale-complementary feature extraction blocks (see Section 7.1.2). Additionally, the fusion process is guided by a a set of scale-aware ground-truth density maps (see Section 7.1.3), resulting in scale-aware features.

Furthermore, we propose a scale complementary feature extraction block (SCFB) which uses cross-scale residual blocks to extract features from adjacent scales in such a way that they are complementary to each other. Traditional fusion approaches such as feature addition or concatenation are not necessarily optimal because they simple merge the features and have limited abilities to extract relevant information from different layers. In contrast, the proposed scale complementary extraction enables the network to compute relevant features from each scale.

Lastly, we address the issue of missing scale-information in crowd-datasets by approximating the same based on the crowd-density levels and superpixel segmentation principles. Zhang *et al.* [1] also estimate the scale information, however, they rely on heuristics based on the nearest number of heads. In contrast, we combine information from the annotations and super-pixel segmentation of the input image in a Markov Random Field (MRF) framework [188].

7.1 Proposed method

In this section, we discuss details of the proposed multi-level feature fusion scheme along with the scale complementary feature extraction blocks. This is followed by a discussion on the estimation of head sizes using the MRF

framework.

7.1.1 Multi-level bottom-top and top-bottom Fusion (MBT-TBF)

The proposed method for crowd counting is based on the recently popular density map estimation approach [18, 60, 76], where the network takes image as an input, processes it and produces a density map. This density map indicates the per-pixel count of people in the image. The network weights are learned by optimizing the L_2 error between the predicted density map and the ground truth density map. As discussed earlier, crowd counting datasets provide x, y locations and these are used to create the ground-truth density maps for training by imposing 2D Gaussians at these locations:

$$D_i(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma), \tag{7.1}$$

where σ is the Gaussian kernel's scale and S is the list of all locations of people. Integrating the density map over its width and height produces the total count of people in the input image.

Fig 7.2 illustrates the overview of the proposed network. We use VGG16 [140] as the backbone network. Conv1 - conv5 in Figure 7.2 are the first five convolutional layers of the VGG16 network. The last layer conv6 is defined as $\{M_2 - C_{512,128,1} - R\}$[1]). As it can be observed from this figure, the network consists of primarily three branches: (i) main branch (VGG16 backbone), (ii)

[1] M_s denotes max-pooling with stride s, $C_{N_i,N_o,k}$ is convolutional layer (where N_i = number of input channels, N_o = number of output channels, $k \times k$ = size of filter), R is activation function (ReLU).

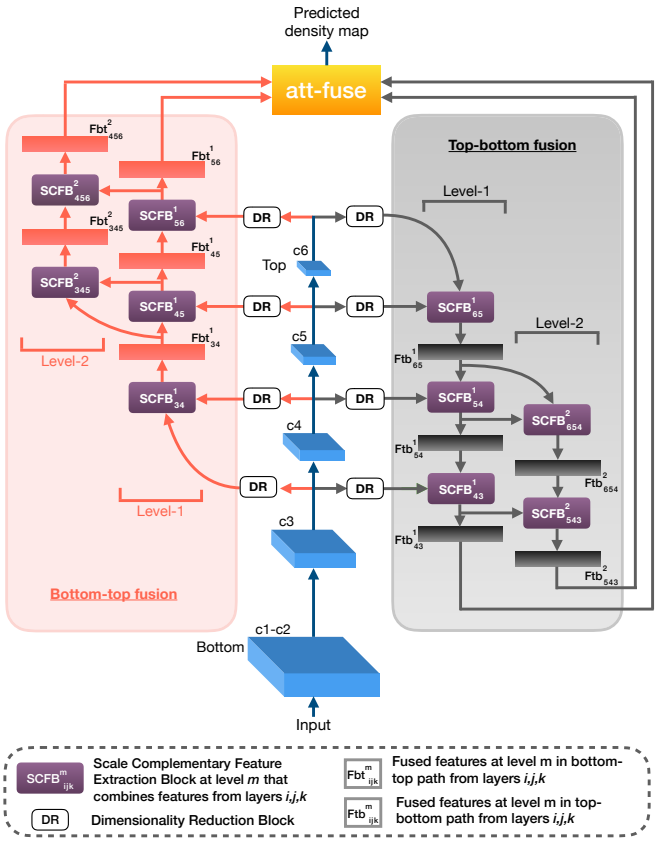


Figure 7.2: Overview of the proposed multi-level top-bottom and bottom-top fusion method for crowd counting.

multi-level bottom-top fusion branch, and (iii) multi-level top-bottom fusion branch. The input image is passed through the main branch and multi-scale features from conv3-conv6 layers are extracted. These multi-scale features are then forwarded through dimensionality reduction (DR) blocks that consists of 1×1 conv layers to reduce the channel dimensions to 32.

The feature maps extracted from the lower conv layers of the main branch contain detailed spatial information which are important for accurate localization, whereas the feature maps from higher layers contain global context and high-level information. The information contained in these different layers are

fused with each other in two separate fusion branches: multi-level bottom-top branch and multi-level top-bottom branch.

Multi-level bottom-top fusion: The bottom-top branch hierarchically propagates spatial information from the bottom layers to the top layers. This branch has two levels of fusion. In the first level, features from the main branch are progressively forwarded through a series of scale complementary feature extraction blocks ($SCFB^1_{34}$-$SCFB^1_{45}$-$SCFB^1_{56}$). First, $SCFB^1_{34}$ combines the feature maps from conv3 and conv4 to produce enriched feature maps Fbt^1_{34}. These features are then combined with conv5 features of the main branch through $SCFB^1_{45}$ to produce Fbt^1_{45}. Finally, these feature maps are combined with conv6 feature maps through $SCFB^1_{56}$ to produce Fbt^1_{56}.

Further, we add another level of bottom-top fusion path which progressively combines features from the first level through another series of scale complementary feature extraction blocks ($SCFB^2_{345}$-$SCFB^2_{456}$). Specifically, Fbt^1_{34} and Fbt^1_{45} are combined through $SCFB^2_{345}$ to produce Fbt^2_{345}. Finally, Fbt^2_{345} is combined with Fbt^1_{56} through $SCFB^2_{456}$ to produce Fbt^2_{456}. The two levels of fusion together form a hierarchy of fusion paths.

Multi-level top-bottom fusion: The bottom-top branch while propagating spatial information to the top layers, inadvertently passes noise information as well. To overcome this, we add a top-bottom fusion path that hierarchically propagates high-level context information into the lower layers. Similar to the bottom-top path, the top-bottom path also consists of two levels of fusion.

In the first level, features from the main branch are progressively forwarded through a series of scale complementary feature extraction blocks ($SCFB^1_{65}$-$SCFB^1_{54}$-$SCFB^1_{43}$). First, $SCFB^1_{65}$ combines the feature maps from conv6 and conv5 to produce enriched feature maps Ftb^1_{65}. These features are then combined with conv4 features of the main branch through $SCFB^1_{54}$ to produce Ftb^1_{54}. Finally, these feature maps are combined with conv3 feature maps through $SCFB^1_{43}$ to produce Ftb^1_{43}.

The second level of bottom-top fusion path progressively combines features from the first level through another series of scale complementary feature extraction blocks ($SCFB^2_{654}$-$SCFB^2_{543}$). Specifically, Ftb^1_{65} and Ftb^1_{54} are combined through $SCFB^2_{654}$ to produce Ftb^2_{654}. Finally, Ftb^2_{654} is combined with Ftb^1_{43} through $SCFB^2_{543}$ to produce Fbt^2_{543}. Again, the two levels of fusion together form a hierarchy of fusion paths in the top-bottom module.

Self attention-based fusion: The features produced by the bottom-top fusion (Fbt^1_{56} and Fbt^2_{456}), although refined, may contain some unnecessary background clutter. Similarly, the features (Ftb^1_{43} and Ftb^2_{543}) produced by the top-bottom fusion may over suppress the detail information in the lower layers. In order to further suppress the background noise in the bottom-top path and avoid over-suppression of detail information due to the top-bottom path, we introduce a self-attention based fusion module at the end that combines feature maps from the two fusion paths. Given the set of feature maps (Fbt^1_{56} , Fbt^2_{456}, Ftb^1_{43} and Ftb^2_{543}) from the fusion branches, the attention

module concatenates them and forwards them through a set of conv layers ($\{C_{128,16,3} - R - \{C_{16,4,1}\}^2$) and a sigmoid layer to produces an attention maps with four channels, with each channel specifying the importance of the corresponding feature map from the fusion branch. The attention maps are calculated as follows: $A = sigmoid(cat(F^1_{56}, F^2_{456}, F^1_{43}, F^2_{543}))$.

These attention maps are then multiplied element-wise to produce the final feature map: $F_f = A^1 \odot F^1_{56} + A^2 \odot F^2_{456} + A^3 \odot F^1_{43} + A^4 \odot F^2_{543}$, where $\odot$ denotes element-wise multiplication. This self-attention module effectively combines the advantages of the two paths, resulting in more powerful and enriched features. Figure 7.3*(a)* shows the self-attention block used to combine different feature maps. The final features F_f are then forwarded through 1×1 conv layer to produce the density map Y_{pred}.

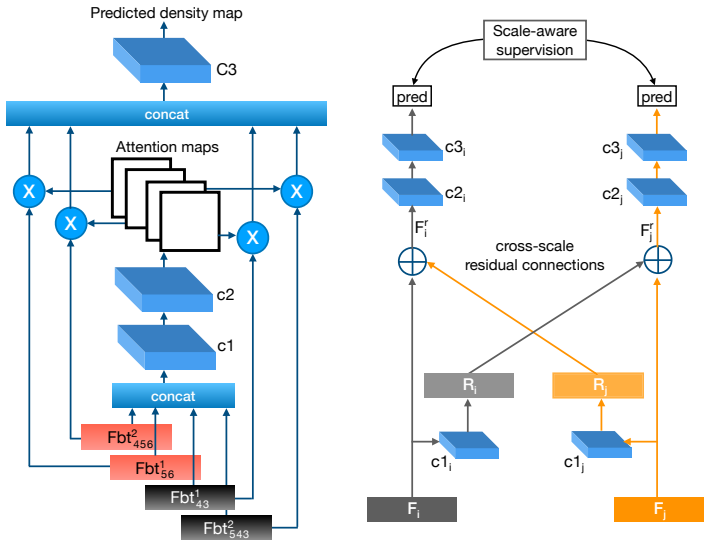


Figure 7.3: (a)Attention fuse module. (b) Scale complementary feature extraction block (SCFB).

Figure 7.4: Scale aware ground truth density maps imposed on the input image. The overall density map is divided into four maps based on the size/scale of the heads. The first image (leftmost) has density corresponding to the smallest set of heads, whereas the last image (rightmost) has densities corresponding to the largest set of heads.

7.1.2 Scale complementary feature extraction block (SCFB)

In this section, we describe the scale complementary feature extraction block that is used to combine features from adjacent layers in the network. Existing methods such as feature addition or concatenation are limited in their abilities to learn complementary features. This is because features of adjacent layers are correlated, and this results in some ambiguity in the fused features. To address this issue, we introduce scale complementary feature extraction block as shown in Figure 7.3*(b)*. This block enables extraction of complementary features from each of the scales being fused. The initial conv layers $c1_i, c1_j, c2_i, c2_j$ in Figure 7.3*(b)* are defined as $\{C_{32,32,3} - R\}^2$, where as the final conv layers $c3_i, c3_j$ are defined as $\{C_{32,1,1} - R\}^2$.

The SCFB consists of cross-scale residual connections (R_i and R_j) which are followed by a set of conv layers. The individual branches in the SCFB are supervised by scale-aware supervision (which is now possible due to the scale estimation framework discussed in Section 7.1.3). More specifically, in order to combine feature maps F_i, F_j from layers i, j, first the corresponding cross-scale residual features F_i^r, F_j^r are estimated and added to the original feature maps F_i, F_j to produce $\hat{F}_i, \hat{F}_j$, *i.e.*, $\hat{F}_i = F_i + F_j^r$ and $\hat{F}_j = F_j + F_i^r$. These features are then forwarded through a set of conv layers, before being supervised by the scale-aware ground-truth density maps Y_i^s, Y_j^s. By adding these intermediate supervisions and introducing the cross-scale residual connections, we are able to compute complementary features from the two scales in the form of residuals. This reduces the ambiguity as compared to the existing fusion methods. For example, if a feature map F_i from a particular layer/scale i is sufficient enough to obtain perfect prediction, then the residual F_j^r is simply driven towards zero. Hence, involving residual functions reduces the ambiguity as compared to the existing fusion techniques.

In order to supervise the SCFBs, we create scale-aware ground-truth density maps based on the scales/sizes estimated as described in Section 7.1.3. Annotations in a particular image are divided into four categories based on the corresponding head sizes, and these four categories are used to create four separate ground-truth density maps ($Y_3^s, Y_4^s, Y_5^s and Y_6^s$) for a particular image. Figure 7.4 shows the four scale-aware ground-truth density maps for two sample images. It can be observed that the first ground-truth (left) has labels corresponding to the smallest heads, where as the last ground-truth (right)

has labels corresponding to the largest heads. These maps ($Y_3^s, Y_4^s, Y_5^s and Y_6^s$) are used to provide intermediate supervision to feature maps coming from conv layers 3,4,5 and 6 coming from the main branch in SCFBs.

7.1.3 Head size estimation using MRF framework

As discussed earlier, the ground truth density maps for training the CNNs are created by imposing 2D Gaussians at the head locations (Equation (7.1)) provided in the dataset. The scale/variance of these Gaussians needs to be decided based on the heads size. Existing methods either assume constant variance [147] or estimate the variance based on the number of nearest heads [1]. Assuming constant variance results in ambiguity in the density maps and hence, prohibits the network to learn scale relevant features. Figure 7.5*(a)* shows the scales for annotations assuming constant variance. On the other hand, estimating the variance based on nearest neighbours leads to better results in regions of high density. However, in regions of low density, the estimates are incorrect leading to ambiguity in such regions (as shown in Figure 7.5*(b)*).

To overcome these issues, we propose a principled way of estimating the scale or variance by considering the input images which were not exploited earlier. We leverage color cues from the input image and combine them with the annotation data to better estimate the scale. Specifically, we first over-segment the input image using a super-pixel algorithm (SLIC [189]) and then combine with watershed segmentation [190] resulting from the distance transform of the head locations in an MRF framework. The size of

the segments resulting from this procedure are then used to estimate the scale of the corresponding head lying in that segment. Figure 7.5*(c)* shows the scales/variances estimated using the proposed method. It can be observed that this method performs better in both sparse and dense regions.

(a) (b) (c)

Figure 7.5: Scale estimation comparison. Scale estimated using (a) Constant scale (b) Nearest neighbours (c) Our method.

7.2 Details of implmentation and training

The network weights are optimized in and end-to-end fashion. We use Adam optimizer with a learning rate of 0.00005 and a momentum of 0.9. We add random noise and perform random flipping of images for data augmentation.Supervision is provided to the network at the final level as well as at intermediate levels in the SCFBs using Euclidean loss. At the final level, the network is supervised by the overall density map (consisting of annotations corresponding to all the heads), whereas the paths in the SCFBs are supervised by the corresponding scale-aware ground-truths.

7.3 Experiments and results

In this section, we first analyze the different components involved in the proposed network through an ablation study. This is followed by a detailed evaluation of the proposed method and comparison with several recent state-of-the-art methods.

7.3.1 Datasets

We use three different congested crowd scene datasets (ShanghaiTech [1], UCF_CROWD_50 [6] and UCF-QNRF [2]) for evaluating the proposed method.

7.3.2 Ablation Study

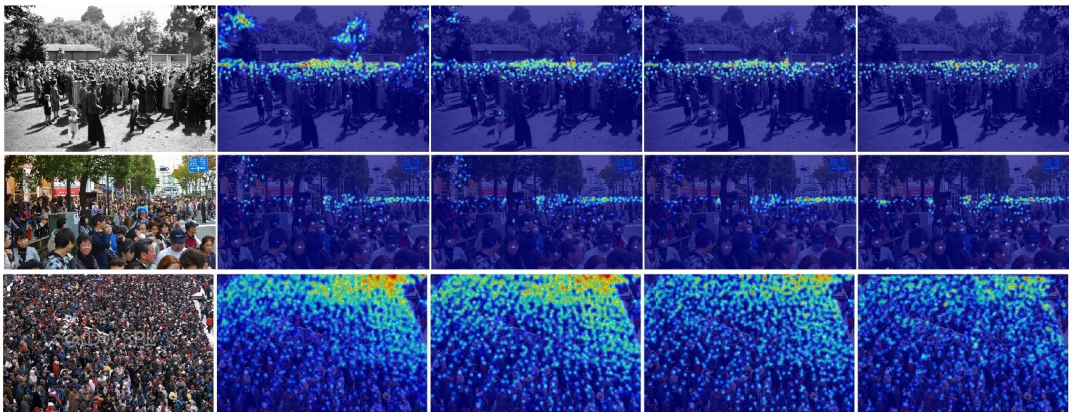
We perform a detailed ablation study to understand the effectiveness of various fusion approaches described earlier. The ShanghaiTech Part A and UCF-QNRF datasets contain different conditions such as high variability in scale, occluded objects and large crowds, *etc*. Hence, we used these datasets for conducting the ablations. The following configurations were trained and evaluated:

(i) *Baseline*: VGG16 network with *conv*6 at the end (Figure 7.1(a)),

(ii) *Baseline* + *fuse-a*: Baseline network with multi-scale feature fusion using feature addition (Figure 7.1(b)),

(iii) *Baseline* + *fuse-c*: Baseline network with multi-scale feature fusion using feature concatenation (Figure 7.1(b)),

(iv) *Baseline* + *BT* + *fuse-c*: Baseline network with bottom-top multi-scale feature fusion using feature concatenation (Figure 7.1(c)),



(a) (b) (c) (d) (e)

Figure 7.6: Ablation study results: (a) Input, (b) Simple feature concatenation (experiment-ii), (c) Bottom-top and top-bottom fusion (experiment - vi), (d) MBTTF (experiment - viii), (e) Ground-truth density map.

(v) *Baseline + TB + fuse-c*: Baseline network with top-bottom multi-scale feature fusion using feature concatenation (Figure 7.1(d)),

(vi) *Baseline + BTTB + fuse-c*: Baseline network with bottom-top and top-bottom multi-scale feature fusion using feature concatenation (Figure 7.1(e)),

(vii) *Baseline + MBTTB + fuse-c*: Baseline network with multi-level bottom-top and top-bottom multi-scale feature fusion using feature concatenation (Figure 7.1(f)),

(viii) *Baseline + MBTTB + SCFB-NS*: Baseline network with multi-level bottom-top and top-bottom multi-scale feature fusion using SCFB, without using scale-aware supervision (Figure 7.2)

(ix) *Baseline + MBTTB + SCFB*: Baseline network with multi-level bottom-top and top-bottom multi-scale feature fusion using SCFB (Figure 7.2)

The quantitative results of the ablation study are shown in Table 7.1. As

Table 7.1: Ablation study results.

Dataset	Shanghaitech-A[1]		UCF-QNRF[2]	
Method	MAE	MSE	MAE	MSE
Baseline (Figure 7.1a)	78.3	126.6	150.2	220.1
Baseline + fuse-a (Figure 7.1b)	73.6	118.4	140.3	210.8
Baseline + fuse-c (Figure 7.1b)	73.4	115.6	135.2	200.2
Baseline + BT + fuse-c (Figure 7.1c)	68.1	122.2	114.1	185.2
Baseline + TB + fuse-c (Figure 7.1d)	70.2	118.5	120.1	188.1
Baseline + BTTB + fuse-c (Figure 7.1e)	66.9	112.2	115.4	174.5
Baseline + MBTTB + fuse-c (Figure 7.1f)	63.2	108.5	105.5	169.5
Baseline + MBTTB + SCFB-NS (Figure 7.2)	62.5	105.1	102.1	168.1
Baseline + MBTTB + SCFB (Figure 7.2)	60.2	94.1	97.5	165.2

it can be observed, simple fusion scheme of addition/concatenation (experiments (i) and (ii)) of multi-scale features at the end, does not yield significant improvements as compared to the baseline network. This is due to the reason that in case of feature fusion at the end, the supervision directly affects the initial conv layers in the main branch, which may not be necessarily optimal.

However, when the features are fused in either bottom-top/top-bottom fashion, the results improve considerably, when compared to the baseline. Since this kind of fusion sequentially propagates the information in a particular direction, the initial conv layers do not get affected directly. The bottom-top and top-bottom (experiment (vi)) further improves the performance. The multi-level bottom-top and top-bottom configuration, in which an additional level of bottom-top and top-bottom fusion path is added (experiment-vii), reduces the count error further, signifying the importance of the multi-level fusion paths.

Next, we replace the fusion blocks in experiment-vii with the SCFB blocks, which amounts to the proposed method as shown in Figure 7.2 (experiment

viii). However, the SCFB blocks are not supervised by the scale-aware ground-truths. The use of these blocks enables the network to propagate relevant and complementary features along the fusion paths, thus leading to improved performance. Finally, we provide scale-aware ground-truth as supervision signal to the SCFB blocks (experiment - ix), which results in further improvements as compared to without scale-aware supervision.

Figure 7.6 shows qualitative results for different fusion configurations. Due to space constraints and also to explain better, we show the results of experiments (iii) *Baseline + fuse-c*, (vi) *Baseline + BTTB + fuse-c*, (ix) *Baseline + MBTTB + SCFB* only. It can be observed from Figure 7.6*(b)*, that simple concatenation of feature maps results in lot of background noise and loss of details in the final predicted density map, indicating that such an approach is not effective. The bottom-top and top-bottom approach, shown in Figure 7.6*(c)* results in the refined density maps, however, they still contain some amount of noise and loss of details. Lastly, the results of experiment (ix) as shown in Figure 7.6*(d)* which have more details where necessary with much lesser background clutter as compared to earlier configurations.

7.3.3 Comparison with recent methods

In this section, we present the results of the proposed method and compare them with several recent approaches on the three different datasets described in Section 7.3.1.

Comparison of results the ShanghaiTech and UCF_CROWD_50 datasets are presented in Table 7.2 and 7.3 respectively. The proposed method achieves

the best results among all the existing methods on the ShanghaiTech Part A dataset and the UCF_CROWD_50 dataset. On the ShanghaiTech B dataset and UCF_CROWD_50 dataset, our method achieves a close 2^{nd} position, only behind CAN [121].

Table 7.2: Comparison of results on ShanghaiTech [1].

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Switching-CNN [104] (CVPR-17)	90.4	135.0	21.6	33.4
TDF-CNN [160] (AAAI-18)	97.5	145.1	20.7	32.8
CP-CNN [147] (ICCV-17)	73.6	106.4	20.1	30.1
IG-CNN [108] (CVPR-18)	72.5	118.2	13.6	21.1
Liu *et al.* [111] (CVPR-18)	73.6	112.0	13.7	21.4
CSRNet [161] (CVPR-18)	68.2	115.0	10.6	16.0
SA-Net [107] (ECCV-18)	67.0	104.5	8.4	13.6
ic-CNN [112] (ECCV-18)	69.8	117.3	10.7	16.0
ADCrowdNet [113] (CVPR-19)	63.2	98.9	8.2	15.7
RReg [114] (CVPR-19)	63.1	96.2	8.7	13.5
CAN [121] (CVPR-19)	**61.3**	100.0	**7.8**	**12.2**
Jian *et al.* [119] (CVPR-19)	64.2	109.1	8.2	**12.8**
HA-CCN [117] (TIP-19)	62.9	**94.9**	8.1	13.4
MBTTBF-SCFB (proposed)	**60.2**	**94.1**	**8.0**	15.5

Results on the recently released large-scale UCF-QNRF [2] dataset are shown in Table 7.4. We compare our results with several recent approaches. The proposed achieves the best results as compared to other recent methods on this complex dataset, thus demonstrating the significance of the proposed multi-level fusion method.

Qualitative results for sample images from the ShanghaiTech dataset are presented in Figure 7.7.

Table 7.3: Comparison of results on UCF_CROWD_50 [3].

	UCF_CROWD_50	
Method	MAE	MSE
Switching-CNN [104] (CVPR-17)	318.1	439.2
TDF-CNN [160] (AAAI-18)	354.7	491.4
CP-CNN [147] (ICCV-17)	295.8	320.9
IG-CNN [108] (CVPR-18)	291.4	349.4
D-ConvNet [110] (CVPR-18)	288.4	404.7
Liu *et al.* [111] (CVPR-18)	289.6	408.0
CSRNet [161] (CVPR-18)	266.1	397.5
ic-CNN [112] (ECCV-18)	260.9	365.5
SA-Net-patch [107] (ECCV-18)	258.5	334.9
ADCrowdNet [113] (CVPR-19)	266.4	358.0
CAN [121] (CVPR-19)	**212.2**	**243.7**
Jian *et al.* [119] (CVPR-19)	249.9	354.5
HA-CCN [117] (TIP-19)	256.2	348.4
MBTTBF-SCFB (ours)	**233.1**	**300.9**

Table 7.4: Comparison of results on the UCF-QNRF datastet [2].

Method	MAE	MSE
CMTL [103] (AVSS-17)	252.0	514.0
MCNN [1] (CVPR-16)	277.0	426.0
Switching-CNN [104] (CVPR-17)	228.0	445.0
Idrees *et al.* [2] (ECCV-18)	132.0	191.0
Jian *et al.* [119] (CVPR-19)	113.0	188.0
CAN [121] (CVPR-19)	107.0	183.0
HA-CCN [117] (TIP-19)	118.1	180.4
MBTTBF-SCFB (ours)	**97.5**	**165.2**

7.4 Summary

We presented a multi-level bottom-top and top-bottom fusion scheme for overcoming the issues of scale variation that adversely affects crowd counting in congested scenes. The proposed method first extracts a set of scale-complementary features from adjacent layers before propagating them hierarchically in bottom-top and top-bottom fashion. This results in a more effective fusion of features from multiple layers of the backbone network. The effectiveness of the proposed fusion scheme is further enhanced by using ground-truth

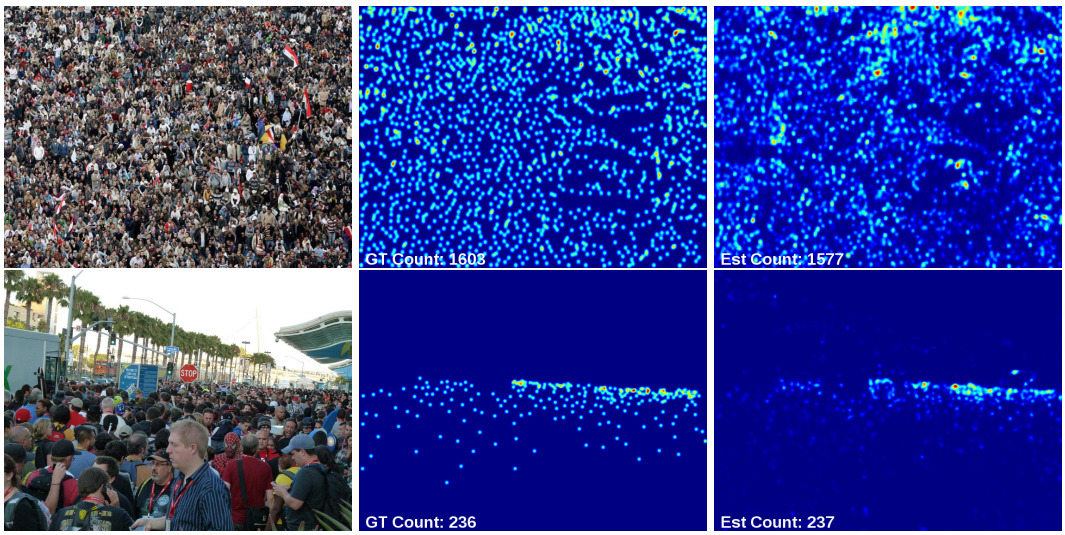


Figure 7.7: Qualitative results of the proposed method on ShanghaiTech [1] *First column:* Input. *Second column:* Ground truth *Third column:* Predicted density map.

density maps that are created in a principled way by combining information from the image and location annotations in the dataset. In comparison to existing fusion schemes and state-of-the-art counting methods, the proposed approach is able to achieve significant improvements when evaluated on three popular crowd counting datasets.

Chapter 8

Confidence Guided Deep Residual Counting Network (CG-DRCN)

In this chapter, we consider the design of network architecture for the task of counting. Design of novel networks specifically for the task of counting has improved the counting error by large margins. Architectures have evolved from the simple ones like [59] which consisted of a set of convolutional and fully connected layers, to the most recent complex architectures like SA-Net [107] which consists of a set of scale aggregation modules. Typically, most existing works ([59, 1, 17, 20, 104, 103, 112, 107, 108, 147, 107, 117]) have designed their networks by laying a strong emphasis on addressing large variations of scale in crowd images. While this strategy of developing robustness towards scale changes has resulted in significant performance gains, it is nevertheless important to exploit other properties like in [112, 109, 110] to further the improvements.

In a similar attempt, we exploit residual learning mechanism for the purpose of improving crowd counting. Specifically, we present a novel design

based on the VGG16 network [140], and it employs residual learning to progressively generate better quality crowd density maps. This use of residual learning is inspired by its success in several other tasks like super-resolution [191, 192, 193, 194, 192]. Although this technique results in improvements in performance, it is important to ensure that only highly confident residuals are used in order to ensure the effectiveness of residual learning. To address this issue, we draw inspiration from the success of uncertainty-based learning mechanism [195, 196, 197]. We propose an uncertainty-based confidence weighting module that captures high-confidence regions in the feature maps to focus on during the residual learning. The confidence weights ensure that only highly confident residuals get propagated to the output, thereby increasing the effectiveness of the residual learning mechanism. Furthermore, we exploit the additional image-level labels in the proposed dataset to extend the uncertainty-based confidence weighting module by conditioning it on the labels to improve the performance specifically in the adverse weather conditions.

8.1 Proposed method

In this section, we present the details of the proposed Confidence Guided Deep Residual Crowd Counting (CG-DRCN) along with the training and inference specifics. Figure 8.1 shows the architecture of the proposed network.

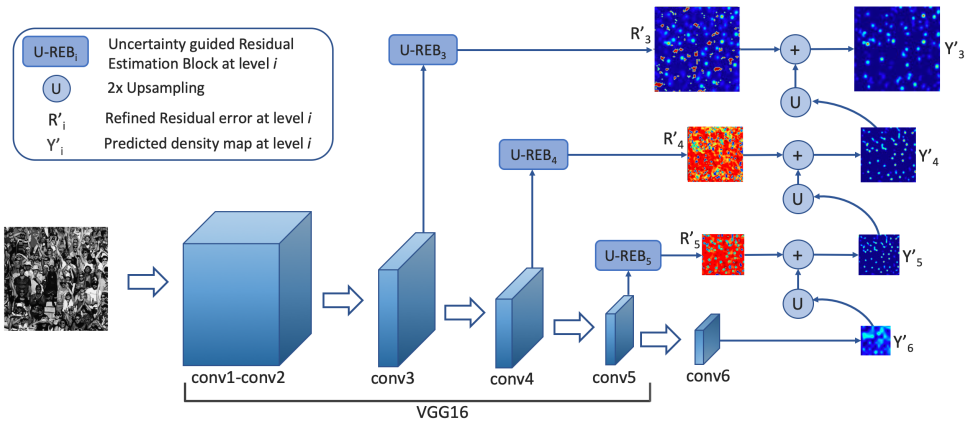


Figure 8.1: Overview of the proposed method. Coarse density map from the deepest layer of the base network is refined using the residual map estimated by the shallower layer. The residual estimation is performed by $U\text{-}REB_i$. In the residual maps, red indicates negative values and cyan indicates positive value.

8.1.1 Base network

Following recent approaches [147, 104, 107], we perform counting based on the density estimation framework. In this framework, the network is trained to estimate the density map ($\hat{Y}$) from an input crowd image (X). The target density map (Y) for training the network is generated by imposing normalized 2D Gaussian at head locations provided by the dataset annotations:

$$Y(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma), \tag{8.1}$$

where, S is the set of all head locations (x_g) in the input image and σ is scale parameter of 2D Gaussian kernel. Due to this formulation, the density map contains per-pixel density information of the scene, which when integrated results in the count of people in the image.

The proposed network consists of conv1$\sim$conv5 layers ($C_1 - C_5$) of the VGG16 architecture as a part of the backbone, followed by a conv block

(CB_6) and a max-pooling layer with stride 2. First, the input image (of size $W \times H$) is passed through $C_1 - C_5$, CB_6 and the max pooling layer to produce the corresponding density map ($\hat{Y}_6$) of size $\frac{W}{32} \times \frac{H}{32}$. CB_6 is defined by {conv$_{512,32,1}$-relu-conv$_{32,32,3}$-relu-conv$_{32,1,3}$}[1]). Due to its low resolution, ($\hat{Y}_6$) can be considered as a coarse estimation, and learning this will implicitly incorporate global context in the image due the large receptive field at the deepest layer in the network.

8.1.2 Residual learning

Although $\hat{Y}_6$ provides a good estimate of the number of people in the image, the density map lacks several local details as shown in Figure 8.3 (a). This is because deeper layers learn to capture abstract concepts and tend to lose low level details in the image. On the other hand, the shallower layers have relatively more detailed local information as compared to their deeper counterparts [134]. Based on this observation, we propose to refine the coarser density maps by employing shallower layers in a residual learning framework. This refinement mechanism is inspired in part by several leading work on super-resolution [191, 192, 193] that incorporate residual learning to learn finer details required to generate a high quality super-resolved image. Specifically, features from C_5 are forwarded through a uncertainty guided residual estimation block ($U\text{-}REB_5$ to generate a residual map $\hat{R}_5$, which is then added to an appropriately up-sampled version of $\hat{Y}_6$ to produce the density map $\hat{Y}_5$

[1] $conv_{N_i,N_o,k}$ denotes conv layer (with N_i input channels, N_o output channels, $k \times k$ filter size), *relu* denotes ReLU activation

of size $\frac{W}{16} \times \frac{H}{16}$, *i.e.*,

$$\hat{Y}_5 = \hat{R}_5 + up(\hat{Y}_6). \tag{8.2}$$

Here, $up()$ denotes up-sampling by a factor of $2\times$ via bilinear interpolation. By enforcing *U-REB*$_5$ to learn a residual map, the network focuses on the local errors emanating from the deeper layer, resulting in better learning of the offsets required to refined the coarser density map. *U-REB* is described in Section 8.1.3.

The above refinement is further repeated to recursively generate finer density maps $\hat{Y}_4$ and $\hat{Y}_3$ using the feature maps from the shallower layers C_4 and C_3, respectively. Specifically, the output of C_4 and C_3 are forwarded through *U-REB*$_4$, *U-REB*$_3$ to learn residual maps $\hat{R}_4$ and $\hat{R}_3$, which are then added to the appropriately up-sampled versions of the coarser maps $\hat{Y}_5$ and $\hat{Y}_4$ to produce $\hat{Y}_4$ and $\hat{Y}_3$ respectively in that order. Specifically, $\hat{Y}_4$ and $\hat{Y}_3$ are obtained as follows:

$$\hat{Y}_4 = \hat{R}_4 + up(\hat{Y}_5), \qquad \hat{Y}_3 = \hat{R}_3 + up(\hat{Y}_4) \tag{8.3}$$

8.1.3 Uncertainty guided residual learning (*U-REB*)

In this section, we provide a detailed description of the uncertainty guided residual estimation block (*U-REB*) that is used to refine the residual estimation process. Specifically, features (F_i) from the main branch are forwarded through a conv block (CB_i) which estimates the residual map R_i. In order to improve the efficacy of the residual learning mechanism, we propose an uncertainty guided confidence estimation block (CEB) to guide the refinement

process. The task of conv blocks CB_i is to capture residual errors that can be incorporated into the coarser density maps to produce high quality density maps in the end. For this purpose, these conv blocks employ feature maps from shallower conv layers C_i.

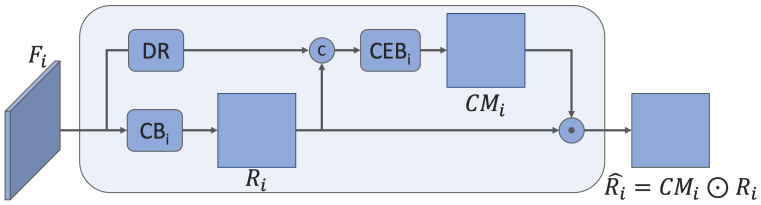
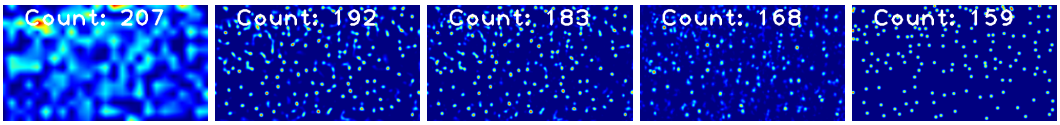


Figure 8.2: Uncertainty-guided residual estimation block (*U-REB*).



(a) (b) (c) (d) (e)

Figure 8.3: Density maps estimated by different layers of the proposed network. (a) $\hat{Y}_6$ (b) $\hat{Y}_5$ (c) $\hat{Y}_4$ (d) $\hat{Y}_3$ (e) Y(ground-truth). It can be observed that the output of the deepest layer ($\hat{Y}_6$) looks very coarse, and it is refined in a progressive manner using the residual learned by *U-REB*$_5$, *U-REB*$_4$, *U-REB*$_3$ to obtain the $\hat{Y}_5$, $\hat{Y}_4$, $\hat{Y}_3$ respectively. Note that fine details and the total count in the density maps improve as we move from $\hat{Y}_6$ to $\hat{Y}_3$.

Since the conv layers in the main branch are primarily trained for estimating the coarsest density map, their features have high responses in regions where crowd is present, and hence, they may not necessarily produce effective residuals. In order to overcome this issue, we propose to gate the residuals that are not effective using uncertainty estimation. Inspired by uncertainty estimation in CNNs [195, 196, 197, 181], we aim to model pixel-wise aleatoric uncertainty of the residuals estimated by CB_i. That is we, predict the pixel-wise confidence (inverse of the uncertainties) of the residuals which are then

used to gate the residuals before being passed on to the subsequent outputs. This ensures that only highly confident residuals get propagated to the output. Note that CB_5, CB_4, CB_3 are defined as follows:

CB_5: $\{\text{conv}_{512,32,1}\text{-relu-conv}_{32,32,3}\text{-relu-conv}_{32,1,3}\}^2$.
CB_4: $\{\text{conv}_{512,32,1}\text{-relu-conv}_{32,32,3}\text{-relu-conv}_{32,1,3}\}^2$.
CB_3: $\{\text{conv}_{256,32,1}\text{-relu-conv}_{32,32,3}\text{-relu-conv}_{32,1,3}\}^2$.

In terms of the overall architecture, we introduce a set of *U-REB*s as shown in Figure 8.1. Each residual branch consists of one such block. Figure 8.2 illustrates the mechanism of the proposed *U-REB*. UREB_i takes the residual F_i from the main branch and forwards them through a conv block CB_i to produce residual map (R_i). This residual map is then concatenated with dimensionality reduced features[2] from the main branch and forwarded through confidence estimation block (CEB_i). This block is defined by $\{\text{conv}_{33,32,1}\text{-relu-conv}_{32,16,3}\text{-relu-conv}_{16,16,3}\text{-relu-conv}_{16,1,1}\}$ and it produces a confidence map CM_i which is then multiplied element-wise with the input to form the refined residual map:

$$\hat{R}_i = R_i \odot CM_i, \tag{8.4}$$

where $\odot$ denotes element-wise multiplication.

In order to learn these confidence maps, the loss function L_f used to train

[2]We use 1×1 conv layer to reduce to 32 channels

the network is defined as follows,

$$L_f = L_d - \lambda_c L_c, \tag{8.5}$$

where, λ_c is a regularization constant, L_d is the pixel-wise regression loss to minimize the density map prediction error and is defined as:

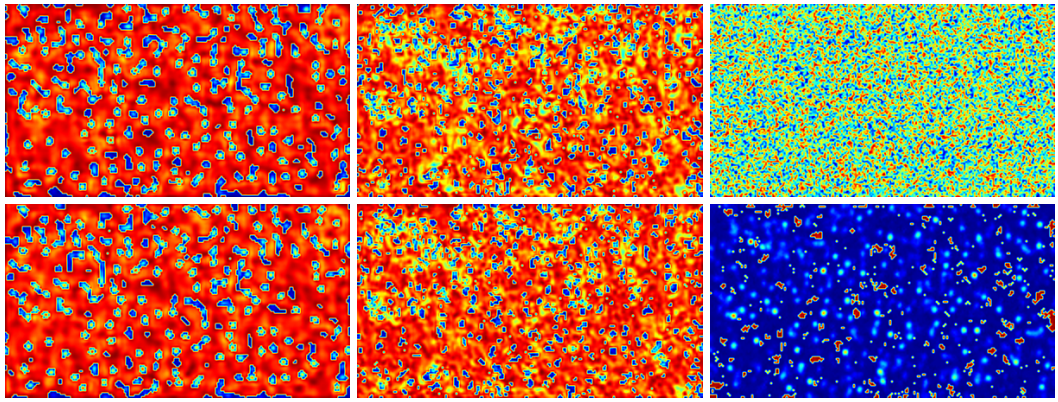
$$L_d = \sum_{i \in \{3,4,5,6\}} \|(CM_i \odot Y_i) - (CM_i \odot \hat{Y}_i)\|_2, \tag{8.6}$$

where, $\hat{Y}_i$ is the predicted density map, i indicates the index of the conv layer from which the predicted density map is taken, Y_i is the corresponding target.

L_c is the confidence guiding loss, defined as,

$$L_c = \sum_{i \in \{3,4,5,6\}} \sum_{j=1}^{H} \sum_{k=1}^{W} \log(CM_i^{j,k}), \tag{8.7}$$

where, $W \times H$ is the dimension of the confidence map CM_i. As it can be



(a) (b) (c)

Figure 8.4: Residual maps. *Top row*: Without confidence gating. *Bottom row*: With confidence gating. (a) R_5 (b) R_4 (c) R_3. Red indicates negative values and cyan indicates positive values. The use of confidence gating improves the residual maps significantly, especially for the shallower layers.

seen from Equation (8.5), the loss L_f has two parts L_d and L_c. The first term minimizes the Euclidean distance between the prediction and target features, whereas L_c maximizes the confidence scores CM_i by making them closer to 1.

Figure 8.3 illustrates the output density maps ($\hat{Y}_6$, $\hat{Y}_5$, $\hat{Y}_4$, $\hat{Y}_3$) generated by the proposed network for a sample crowd image. It can be observed that the density maps progressively improve in terms of fine details and the count value.

Figure 8.4 illustrates the residual maps generated with and without the confidence gating. It can be clearly observed that the use of confidence scores aids in better feature learning.

8.1.4 Class-conditioned Uncertainty guided residual learning (*U-REBC*)

In order to leverage additional information provided in the proposed JHU-CROWD++ dataset, we propose to condition the residual estimation based on the image-level labels (specifically, weather labels). That is, we augment the *U-REB* module with additional class conditioning (*CC*) block as shown in Figure 8.5. This block consists of a set of 2 conv relu-layers ($\{\text{conv}_{32,32,3}\text{-relu-conv}_{32,4,3}\}^2$) followed by an average-pool layer and a soft-max layer. Note that the output of this block is 4 classes corresponding to *rain*, *fog*, *haze* and *normal*. The *CC* block is trained via cross-entropy error using labels available in the dataset. To condition the uncertainty estimation on the classes, the feature maps (F_i^c) prior to the average-pool layer in *CC* are concatenated with the residual map R_i and the dimensionality reduced features from the main

Table 8.1: Results of ablation study using **"VGG16"** base network on the JHU-CROWD++ dataset (val-set).

Method	MAE	MSE
Base network	81.1	300.5
Base network + R	77.5	290.6
Base network + R + UREB ($\lambda_c = 0$)	77.1	290.5
Base network + R + UREB ($\lambda_c = 1$)	74.1	275.5
Base network + R + UREB-C ($\lambda_w = 0$)	74.6	274.1
Base network + R + UREB-C ($\lambda_w = 0.01$)	67.9	262.1

branch. These concatenated feature maps are then forwarded through the confidence estimation block CEB_i to predict the confidences as described earlier in Section 8.1.3.

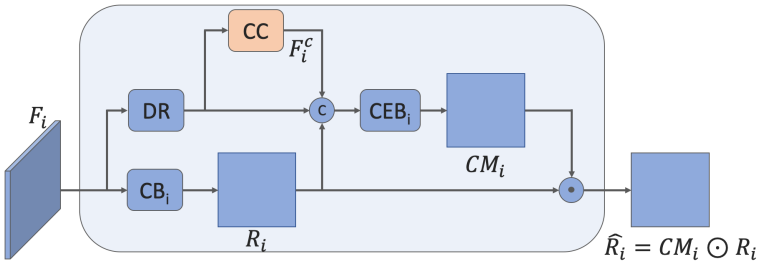


Figure 8.5: Class-conditioned uncertainty-guided residual estimation block (*UREBC*).

For training the network, we modify the loss function in Equation 8.5 as follows:

$$L_f = L_d - \lambda_c L_c + \lambda_w L_w, \tag{8.8}$$

where, L_w is the cross-entropy loss for the weather classification and λ_w is a weighting factor and we set it to 0.01. Note that the distribution of weather images is imbalanced. Hence, we weight the each class proportionately based on the number of samples in each category.

Table 8.2: Ablation results: **"Class-conditioning"** for weather-conditions study on the JHU-CROWD++ weather dataset (val-set).

Method	MAE	MSE
w/o conditioning	78.4	170.5
with conditioning	63.6	116.6

8.1.5 Training and inference details

The training dataset is obtained by cropping patches from multiple random locations in each training image. The cropped patch-size is 256×256. For JHU-CROWD++, we use the validation set for model selection and hyper-parameter tuning. For other datasets, we use 10% of the training images as validation set. We use the Adam optimizer to train the network. We use a learning rate of 0.00001 and a momentum of 0.9 with a batch-size of 24. Before cropping, we resize all the images such that the minimum dimension is 512 and maximum dimension is 2048 while maintaining the aspect ratio.

For inference, the density map $\hat{Y}_3$ is considered as the final output.

8.2 Ablation Study

In this section, we discuss the results of different ablation studies conducted to analyze (i) the effect of different components in the proposed network, (ii) generalizability to other network architectures, and (iii) the effect of different branches in the proposed architecture for residual estimation. Due to the presence of various complexities such as high density crowds, large variations in scales, occlusion, *etc*. we choose to perform the ablation study on JHU-CROWD++ validation set.

8.2.1 Residual Learning, Uncertainty and Class-conditioning

The ablation study consisted of evaluating the following configurations of the proposed method:

(i) *Base network*: VGG16 network with an additional conv block (CB_6) at the end.

(i) *Base network + R*: the base network with residual learning.

(iii) *Base network + R + U-REB ($\lambda_c = 0$)*: the base network with residual learning guided by the confidence estimation blocks as discussed in Section 8.1.3. In this configuration, we aim to measure the performance due to the addition of the confidence estimation blocks without the uncertainty estimation mechanism by setting λ_c is set to 0.

(iv) *Base network + R + U-REB ($\lambda_c = 1$)*: the base network with residual learning guided by the confidence estimation blocks as discussed in Section 8.1.3.

(v) *Base network + R + U-REBC ($\lambda_w = 0$)*: the base network with residual learning guided by the class-conditioned confidence estimation blocks as discussed in Section 8.1.4. In this configuration, we aim to measure the performance due to the addition of conv block in the class conditioning *CC* module without image-level training by setting λ_w is set to 0.

(vi) *Base network + R + U-REBC ($\lambda_w = 0.01$)*: the base network with residual learning guided by the class-conditioned confidence estimation blocks as discussed in Section 8.1.4.

The results of these experiments are shown in Table 8.1. It can be seen that there are considerable improvements in the performance due to the inclusion

Table 8.3: Results of ablation study using **"Res101"** base network on the JHU-CROWD++ dataset (val-set).

Method	MAE	MSE
Base network	72.1	280.5
Base network + R	68.5	270.9
Base network + R + UREB ($\lambda_c = 0$)	68.2	271.2
Base network + R + UREB ($\lambda_c = 1$)	62.5	258.1
Base network + R + UREB-C ($\lambda_w = 0$)	63.1	259.9
Base network + R + UREB-C ($\lambda_w = 0.01$)	57.6	244.4

Table 8.4: Results of ablation on the **"branches"** used for density estimation on the JHU-CROWD++ dataset (val-set).

Base network	VGG16		Res101	
Branch	MAE	MSE	MAE	MSE
$\hat{Y}_6$	81.1	300.5	72.1	280.5
$\hat{Y}_6 + \hat{Y}_5$	72.1	280.1	60.6	251.4
$\hat{Y}_6 + \hat{Y}_5 + \hat{Y}_4$	70.7	270.5	58.8	249.4
$\hat{Y}_6 + \hat{Y}_5 + \hat{Y}_4 + \hat{Y}_3$	67.9	262.1	57.6	244.4

of residual learning into the network. The use of confidence-based weighting of the residuals results in further improvements, thus highlighting its significance in improving the efficacy of uncertainty-based residual learning. Further, as shown in Table 8.2, conditioning the estimation based on the class labels results in improvements specifically for the images captured under adverse weather conditions. This leads to significant improvements in the overall error.

8.2.2 Res101 backbone network

In order to demonstrate that the proposed uncertainty-guided residual learning mechanism is not network-dependent, we evaluate the method using a different base network: Res101 [198]. To employ the Res101 architecture as the base network: we (i) add the uncertainty-based residual estimation blocks

$U\text{-}REB_3$, $U\text{-}REB_4$ and $U\text{-}REB_5$ after layers 2, 3 and 4 in Res101 respectively, (ii) add conv6 layer after layer 5 with the input number of channels changed appropriately to match the number of output channels of layer 4 in Res101, and (iii) change the number of input channels in the conv blocks in $U\text{-}REB_i$'s to match the number of output channels of the respective blocks in the main branch of Res101. Furthermore, since the $U\text{-}REB_3$ is added to a shallower layer, we weight the loss function corresponding to $\hat{Y}_6$. That is, we modify Equation 8.6 as follows:

$$L_d = \sum_{i \in \{3,4,5,6\}} \lambda_i \| (CM_i \odot Y_i) - (CM_i \odot \hat{Y}_i) \|_2. \tag{8.9}$$

In the above equation, we set $\lambda_3 = 0.1$ and $\lambda_4 = \lambda_5 = \lambda_6 = 1$.

Table 8.3 shows the results of the proposed network using Res101 backbone network. We make similar observations as in the case of VGG16 base network. That is, the use of residual learning results in better performance compared to the base network. Further, incorporating uncertainty-guided residual estimation and class conditioning results in further improvements. From this experiment, we can observe that the proposed method can generalize to other types of network architectures.

8.2.3 Number of branches

Since the proposed method involves residual learning at multiple scales of the base network, we conduct a set of experiments to understand the effectiveness of using multiple scales. We evaluate for two backbone architectures: VGG16 and Res101. Specifically, we conduct experiments where we sequentially add

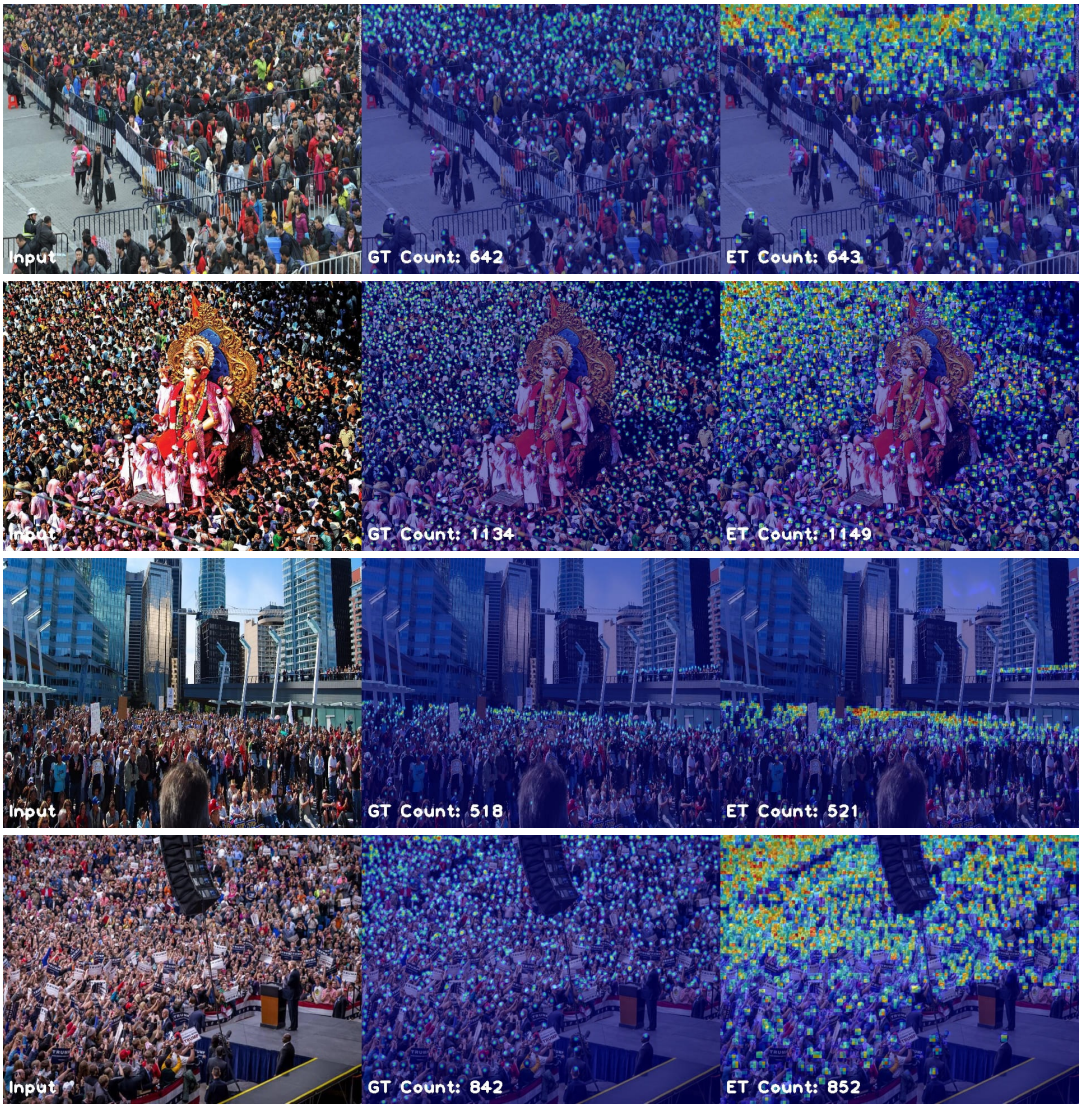
the residual estimation blocks at conv5, conv4 and conv3 for VGG16 and at layer4, layer 3 and layer 2 for Res101. Table 8.4 shows the results of these experiments. It can be observed for both architectures that as we add more residual estimation blocks at different layers, the errors drops by considerable margins.

8.3 Evaluation

In this section, we evaluate the proposed method on datasets like ShanghaiTech [1] and UCF-QNRF [2]. In addition, we compare the proposed method with several recent methods and demonstrate that our method is able to achieve comparable performance with respect to the state-of-the-art methods.

8.3.1 ShanghaiTech Dataset

The proposed network is trained on the train splits using the same strategy as discussed in Section 8.1.5. Table 8.5 shows the results of the proposed method on ShanghaiTech as compared with several recent approaches: CP-CNN[147], IG-CNN [108], D-ConvNet [110], Liu *et al.* [111], CSR-Net [161], ic-CNN [112], SA-Net[107], ACSCP [109] and Jian *et al.* [119], CA-Net [199], BCC [125], DSSI-Net [199], MBTTBF [118] and LSC-CNN [124]. It can be observed that the proposed method outperforms all existing methods on Part A of the dataset, while achieving comparable performance on Part B.



(a) (b) (c)

Figure 8.6: Results of the proposed dataset on sample images from the JHU-CROWD++ dataset. (a) Input image (b) Ground-truth density map (c) Estimated density map.

8.3.2 UCF-QNRF Dataset

Table 8.6 shows results on the UCF-QNRF dataset. The proposed method is compared with the following recent methods: Idrees *et al.* [6], MCNN [1],

Table 8.5: Results on **"ShanghaiTech"** dataset [1].

	Part-A		Part-B	
Method	MAE	MSE	MAE	MSE
CP-CNN [147]	73.6	106.4	20.1	30.1
IG-CNN [108]	72.5	118.2	13.6	21.1
Liu *et al.* [111]	73.6	112.0	13.7	21.4
D-ConvNet [110]	73.5	112.3	18.7	26.0
CSRNet [161]	68.2	115.0	10.6	16.0
ic-CNN [112]	69.8	117.3	10.7	16.0
SA-Net [107]	67.0	104.5	8.4	13.6
ACSCP [109]	75.7	102.7	17.2	27.4
Jian *et al.* [119]	64.2	109.1	8.2	12.8
CA-Net [121]	61.3	100.0	7.8	12.2
BCC [125]	62.8	117.0	8.1	12.7
DSSI-Net [199]	60.6	96.0	**6.8**	**10.3**
MBTTBF [118]	**60.2**	**94.1**	8.0	15.5
LSC-CNN [124]	66.5	101.8	7.7	12.7
CG-DRCN-VGG16 (ours)	64.0	98.4	8.5	14.4
CG-DRCN-Res101 (ours)	**60.2**	**94.0**	**7.5**	**12.1**

CMTL [103], Switching-CNN [104], Idrees *et al.* [2], Jian *et al.* [119], CA-Net [199], BCC [125], DSSI-Net [199], MBTTBF [118] and LSC-CNN [124]. It can be observed that the proposed method achieves comparable performance with respect to the recent state-of-the-art methods.

Table 8.6: Results on **"UCF-QNRF "** dataset [2].

Method	MAE	MSE
Idrees *et al.* [6]	315.0	508.0
Zhang *et al.* [59]	277.0	426.0
CMTL *et al.* [103]	252.0	514.0
Switching-CNN [104]	228.0	445.0
Idrees *et al.* [2]	132.0	191.0
Jian *et al.* [119]	113.0	188.0
CA-Net [121]	107.0	183.0
DSSI-Net [199]	99.1	159.2
MBTTBF [118]	97.5	165.2
BCC [125]	**88.7**	**154.8**
LSC-CNN [124]	120.5	218.2
CG-DRCN-VGG16 (ours)	112.2	176.3
CG-DRCN-Res101 (ours)	**95.5**	**164.3**

8.4 Summary

We presented a novel crowd counting network that employs residual learning mechanism in a progressive fashion to estimate coarse to fine density maps. The efficacy of residual learning is further improved by introducing an uncertainty-based confidence weighting mechanism that is designed to enable the network to propagate only high-confident residuals to the output. Additionally, we incorporate class-conditioning mechanism to leverage the image-level labels in the new dataset for improving the performance in adverse weather conditions. The proposed method is evaluated on recent datasets and we demonstrate that it achieves comparable performance with respect to the state-of-the-art methods.

Chapter 9

Weakly Supervised Crowd Counting

A major issue in the crowd counting research community is the poor generalization performance of the existing networks. This is due to the fact that CNN-based methods are highly data-driven and suffer from inherent dataset bias. Hence, they cannot be applied directly to new scenes without further fine-tuning. A simple solution to this would be to train the model on the target dataset in a fully-supervised fashion, which requires expensive ground-truth annotations. Several earlier works such as [59, 111] address this issue by proposing different semi-supervised or unsupervised fine-tuning methods in addition to their novel network designs. For instance, Zhang *et al.* [59] presented a cross-scene counting approach where they use perspective maps to retrieve candidate scenes from source dataset that are similar to the target set, which are then used to fine-tune the network. However, perspective maps may not be always available. Additionally, it is dependent on the assumption that their pre-trained model provides good estimates of count in the target patches. Liu *et al.* [111] proposed a self-supervised method based on image

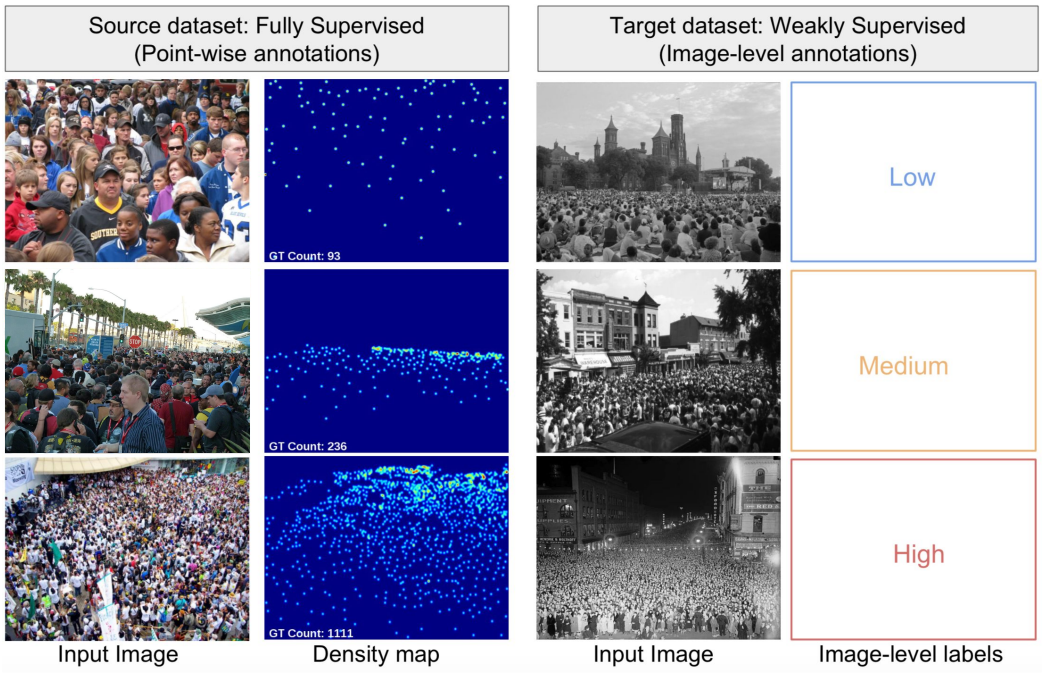


Figure 9.1: Target dataset adaptation. (a) Source dataset with point-wise annotations is used to train the counting network. (b) Target dataset with only image-level annotations is used to fine-tune the pre-trained counting network.

ranking to adapt to different datasets. While it achieves better generalization performance, their method is still limited since they use only unlabeled data.

To address this generalization issue, we take a different approach as compared to earlier attempts ([59, 111]) by proposing a novel weakly supervised learning setup. We leverage image-level labels, which are much easier to obtain as compared to point-wise annotations [1], in a weakly supervised fashion for fine-tuning networks to newer datasets/scenes. To achieve this weak supervision, we use the idea of image-level labeling of crowd images into

[1]Crowd counting datasets are usually provided with point-wise (x,y) location annotations, which are converted to pixel-wise density maps.

different density levels by Sindagi *et al.* [147] and Fu *et al.* [95]. While these methods [147, 95] employ image-level labels in conjunction to point-wise annotations to train their networks, we propose to use only image-level labels in the weakly supervised setup while adapting to new datasets, thereby avoiding the labour intensive point-wise annotation process. Figure 9.1 illustrates the different types of annotations used for training the network. Figure 9.1(a) represents samples from a source dataset, which consists of images and corresponding point-wise ground-truth annotations. The source dataset is used to pre-train the counting network. Figure 9.1(b) represents samples from the target set to which we intend to adapt the pre-trained counting network. The pre-trained network is then fine-tuned on the target dataset using image-level labels via the proposed weakly supervised approach.

9.1 Weak supervision via image-level labels

As discussed earlier, existing methods [1, 59] recognize the inability of these networks to generalize well to different datasets. Their solutions to improve the cross-dataset performance is through fine-tuning in either a fully-supervised or semi-supervised fashion. In contrast to these approaches, we propose a weakly supervised setup to train the counting networks on the new datasets with just image-level labels. Such a setup will simplify the training process as it does not require point-wise annotations which are labour intensive and expensive to obtain.

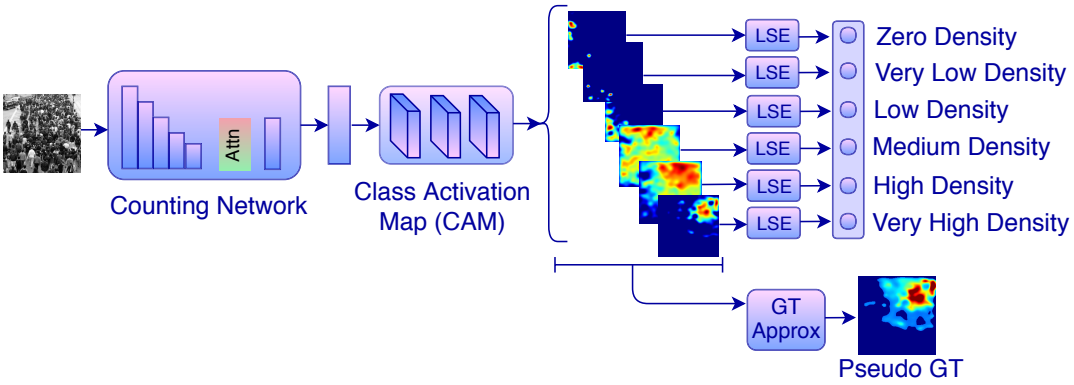


Figure 9.2: Overview of the proposed weakly supervised learning for adapting counting network to new datasets. A class activation map (CAM) module is learned to produce class-wise score maps using image-level labels, which are further used to estimate pseudo ground-truth density maps for target set images.

The idea of performing weakly supervised crowd counting is largely inspired by the success of recent CNN-based weakly supervised semantic segmentation methods [200, 201, 202] that typically fit the problem into Multiple-Instance Learning framework [203]. In their setup, every image is considered to have at least one pixel corresponding to image class label, and the segmentation task is formulated as inferring the pixels belonging to the object class. These methods usually employ class activation mappings to perform weak supervision. However, crowd counting is a regression problem and cannot be directly fit into such a framework. To overcome this issue, crowd counting is transformed into a crowd-density classification task, i.e., instead of counting the number of people in an image, this task is reformulated into categorizing the image into one of the six classes: $\mathcal{C}$ = *{zero density, very-low density, low density, medium density, high density, very-high density}*. This reformulation is based on the intuition that it is easier to label an image as containing large or few number of people as compared to the exact count. Sindagi *et al.* [147]

used a similar concept for leveraging image context. Here, the labels are used to reformulate the counting problem into a classification task for weakly supervised learning.

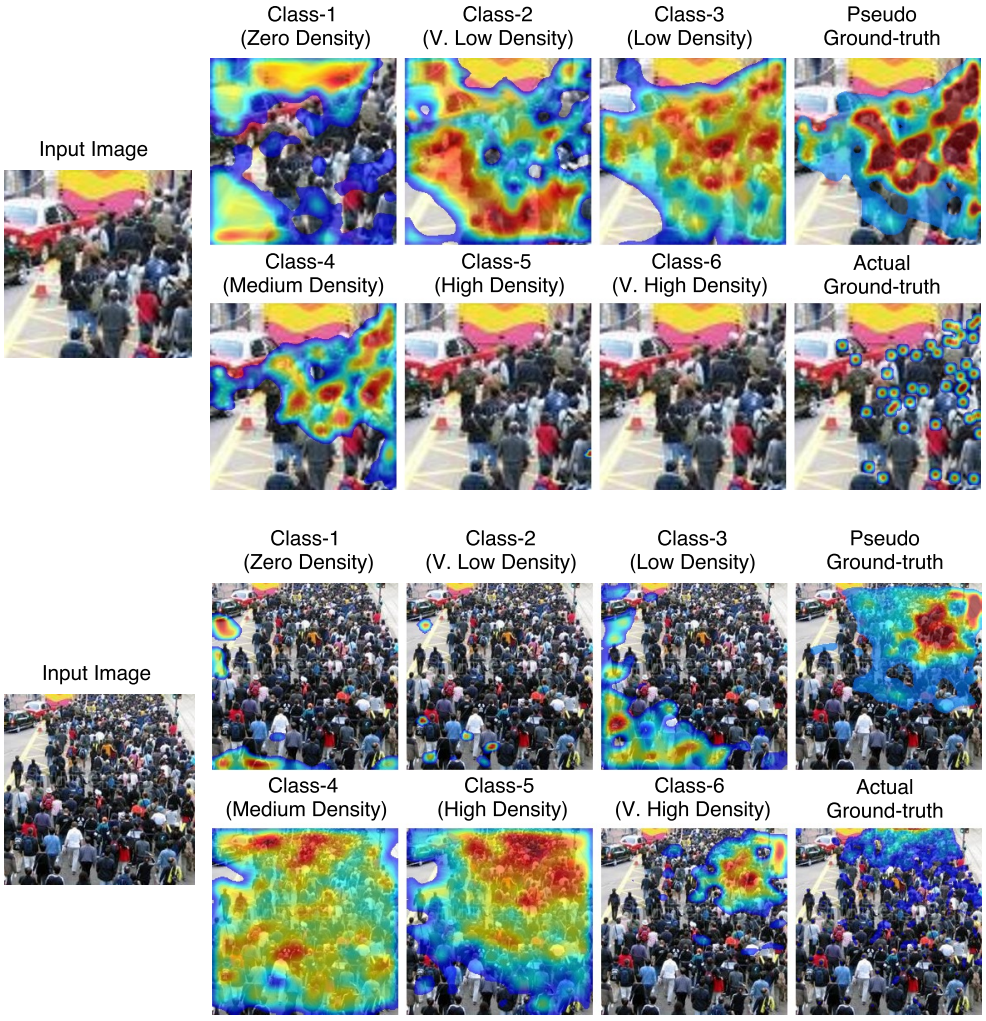


Figure 9.3: Example of class-wise score maps overlaid on input images. It can be observed that the CAM module is able to accurately identify regions corresponding to different density levels in an image. We also illustrate pseudo ground-truth estimated using image-level labels. Note that in the density maps, red color indicates high density and blue color indicates low density.

Figure 9.2 illustrates the proposed weakly supervised approach for adapting to new target scenes or datasets. Similar to semantic segmentation where a pre-trained CNN is used, we use counting network (HA-CCN) described in Chapter 6 that is pre-trained on the source dataset. A class activation map module (CAM), consisting of 4 conv layers is added before the fusion module in the counting network. This module is defined as: {Conv2d(72,64,3)-ReLU, Conv2d(64,64,3)-ReLU, Conv2d(64,32,3)-ReLU}. Conv2d(32,6,3)[2]

This sub-network takes in features from the counting network and processes them to produce output with $|\mathcal{C}|$ feature planes, one for each class. That is, the output of CAM is pixel-wise scores for each class and is denoted by $S^c_{i,j}$ at pixel location (i,j) for each class $c \epsilon \mathcal{C}$. Since point-wise labels are not available for the target set, the pixel-wise scores for each class are mapped to a single image-level classification score using an aggregation function F_{agg} such that $s^c = F_{agg}(S^c_{i,j})$. This class-wise (s^c) score is then maximized for the right class label. Different aggregation functions such as Global Average Pooling (GAP) and Global Max Pooling (GMP) [204] have been used in the literature. In case of GAP, all pixels in the score map are assigned the same weights even if they do not belong to image's class label. GMP addresses this by assigning weight to the pixel that contributes most to the score, however the training is slow [200]. Hence, smooth version and convex approximation of the max function is chosen for F_{agg}, called Log-Sum-Exp (LSE) which is defined as:

$$S^c = \frac{1}{r} \log \left[\frac{1}{wh} \sum_{i,j} \left(r S^c_{i,j} \right) \right], \tag{9.1}$$

[2] *Conv2d(N_i,N_o,k)* denotes 2d convolutional layer (with N_i input channels, N_o output channels, $k \times k$ filter size)

where, S^c denotes aggregated score for class c, $S^c_{i,j}$ is pixel-level score at location (i, j) for class c, r is a hyper-parameter that controls the smoothness of approximation, w, h are width and height of the score map. A soft-max function is applied to the aggregated class scores. The CAM module is trained using the standard binary cross entropy loss function. Parameters of the counting network are kept fixed during this training. The class-wise score maps obtained from the above procedure indicate regions/pixels in the image that belong to a particular density level and hence can be viewed similar to class activation maps [35] (see Figure 9.3). These class-wise maps are then used to approximate the pseudo ground-truth density maps for the target set using:

$$D_{pseudo}(i, j) = \sum_{c \epsilon \mathcal{C}} n(c) \widetilde{S}^c_{i,j}, \tag{9.2}$$

where, $\widetilde{S}^c_{i,j}$ are obtained by normalizing $S^c_{i,j}$ and $n(c)$ is the average count for class c in the source dataset. The pseudo ground-truth maps (as seen in Figure 9.3) are not as sharp as actual ground-truth maps, however, they provide coarse regional density that is better as compared to just image-level labels.

These pseudo ground-truth density maps are used to supervise the counting network on the target dataset. During fine-tuning, weights of the VGG-16 network are fixed and only the weights of the later conv layers are updated. This ensures that the resulting estimated density maps are sharper since the feature maps extracted from VGG-16 preserve details, while the later layers adapt to the newer dataset.

Although the network is trained using image-level labels, it learns to generate density maps for the target set as well. Hence, during inference, test

image from the target set is forwarded through the network to estimate the density map. The performance of the proposed weakly supervised technique is measured using standard count error metrics (MAE/MSE).

9.2 Experiments and results

Table 9.1: Cross dataset performance. S: Model is trained on target set, NS: Model is trained on source and tested on target set. C: Drop in performance between S and NS.

Method	Target Set				
	ShanghaiTech B		UCF_CROWD_50		WExpo '10
	MAE (S/NS/C)	MSE (S/NS/C)	MAE (S/NS/C)	MSE (S/NS/C)	MAE (S/NS/C)
MCNN [1]	26.4/39.6/13.2	41.3/102.5/61.2	377.6/397.7/20.1	509.1/624.1/115.0	11.6/25.2/13.6
Switch CNN [104]	21.6/59.4/37.8	33.4/130.7/97.3	318.1/1117.5/799.4	439.2/1315.4/876.2	9.4/31.1/21.7
D-ConvNet [110]	18.7/49.1/30.4	26.0/99.2/73.2	288.4/364.0/75.6	404.7/545.8/141.1	-
HA-CCN (ours)	8.1/29.1/21.0	13.4/74.1/60.1	256.2/339.8/83.6	348.4/463.2/114.8	8.5/22.0/13.5

We compare the generalization abilities of the proposed method with that of recent methods (MCNN [1], Switching-CNN [104], D-ConvNet [110]) by testing the network (trained on ShanghaiTech A dataset) on target datasets such as ShanghaiTech B, UCF_CROWD_50 and WorldExpo '10 [59]. The results are presented in Table 9.1. Note that the other networks are also trained on ShanghaiTech A dataset. The cross-dataset performance is measured using the overall count error (MAE/MSE) and the drop in performance. The drop in performance is the difference between the error of the model trained on the target set and that of the model trained on source set, when tested on target set. It can be observed that the proposed method is relatively more robust to change in dataset distribution as compared to the other methods.

Although the proposed method demonstrates better cross-dataset performance as compared to existing methods, there is considerable gap in the

performance as compared to when the network is fully supervised on the target set. We address this issue via the weakly supervised technique described in Section 9.1.

9.2.1 Weakly supervised counting

In this section, we present the experiment details and results of weak supervision setup.

Training. First, a source training set is created that is based on the ShanghaiTech A dataset. The other datasets (ShanghaiTech B, UCF_CROWD_50 and WorldExpo) are used as the target sets. ShanghaiTech A is chosen for creating the source training set since it contains large variations in density, scale and appearance of people across images. The training set is created by cropping multi-scale patches of size 224×224 from 9 random locations. The multi-scale patch extractions increases diversity of the source dataset in terms of count and field of view. Image-level labels for the source dataset are assigned based on the count in each image in the source set.

The target training set is created by cropping multi-scale patches from 9 random locations from each image. The image-level labels for the target set are obtained based on the count in each image. To compensate for the fact that count values from the target set are used to obtain the image-level labels (which is not practically feasible since the target set is not supposed to have point-wise or count annotations), label noise is added for 15% of the training samples. That is, we randomly changed the labels of 15% of the samples with the neighboring classes. This process of adding label noise simulates human

labeling error.

The crowd counting network is first trained on the diverse source dataset using full-supervision by minimizing the loss function described in (8.6), followed by addition of the CAM module. Weights of the counting network are fixed and the CAM module is trained on the diverse source dataset by minimizing the binary cross entropy between image-level labels and aggregated class scores. This is followed by fine-tuning of the CAM module on the target samples using image-level labels. The class-wise maps from the CAM module are used to generate the pseudo ground-truth density maps for the target samples which are then used to fine-tune the counting network.

Discussion. The results of adapting pre-trained counting model using weak supervision and selective fine-tuning for three target datasets (ShanghaiTech B, UCF_CROWD_50 and WorldExpo) are shown in Table 9.2. For WorldExpo, we average the MAE error over all the five scenes. For weak supervision, following configurations with three different aggregation functions are evaluated:

(1)HA-CCN+W-A: Global Average Pooling (GAP)

(2)HA-CCN+W-M: Global Max Pooling (GMP)

(3)HA-CCN+W-L: Log-Sum-Exponential (LSE)

It can be observed that the proposed WSL setup results in significant improvements in the generalization performance of the network. Among the three aggregation functions for weakly supervised learning, LSE outperforms the other two functions. The results obtained using WSL are comparable to many recent fully supervised techniques such as Hydra-CNN [20], MCNN [1],

Walach *et al.* [17], Switching-CNN [104], thus demonstrating the significance of the proposed weak supervision technique.

Table 9.2: Results for weakly supervised experiments

Method	Target Set				
	ShanghaiTech B		UCF_CROWD_50		WExpo '10
	MAE	MSE	MAE	MSE	MAE
HA-CCN - NS	29.1	74.1	339.8	463.2	22.0
HA-CCN + W-A	23.1	50.6	320.6	430.6	17.5
HA-CCN + W-M	22.5	51.2	322.2	428.1	17.7
HA-CCN + W-L	21.5	46.1	315.1	420.3	15.9

9.3 Conclusions

We presented a novel weakly supervised setup to adapt counting models to different datasets using image-level labels. Extensive experiments performed on challenging datasets and comparison with recent state-of-the-art approaches demonstrated the significant improvements achieved by the proposed method.

Chapter 10

Learning to Count in the Crowd from Limited Labeled Data

Several convolutional neural network (CNN) based approaches have been developed that address various issues in counting like scale variations, occlusion, background clutter [12, 59, 37, 104, 147, 111, 110, 109, 107, 112, 161, 127, 124, 205, 108], *etc*. While these methods have achieved excellent improvements in terms of the overall error rate, they follow a fully-supervised paradigm and require several labeled data samples. There is a wide variety of scenes and crowded scenarios that these networks need to handle to in the real world. Due to a distribution gap between the training and testing environments, these networks have limited generalization abilities and hence, procuring annotations becomes especially important. However, annotating data for crowd counting typically involves obtaining point-wise annotations at head locations, and this is a labour intensive and expensive process. Hence, it is infeasible to procure annotations for all possible scenarios. Considering this, it is crucial to reduce the annotation efforts, especially for crowd counting methods which get deployed in a wide variety of scenarios.

With the exception of a few works [206, 111, 4], reducing annotation efforts while maintaining good performance is relatively less explored for the task of crowd counting. Hence, we focus on learning to count using limited labeled data while leveraging unlabeled data to improve the performance. Specifically, we propose a Gaussian Process (GP) based iterative learning framework where we augment the existing networks with capabilities to leverage unlabeled data, thereby resulting in overall improvement in the performance. Inspired by [207], the proposed framework follows a pseudo-labeling approach, where we estimate the pseudo-ground truth (pseudo-GT) for the unlabeled data, which is then used to supervise the network. The network is trained iteratively on labeled and unlabeled data.

10.1 Preliminaries

In this section, we briefly review the following concepts: crowd counting, semi-supervised learning and Gaussian Process.

Crowd counting. Following recent works [59, 1], we employ the approach of density estimation technique. That is, an input crowd image is forwarded through the network, and the network outputs a density map. This density map indicates the per-pixel count of people in the image. The count in the image is obtained by integrating over the density map. For training the network using labeled data, the ground-truth density maps are obtained by imposing 2D Gaussians at head location x_g using $D(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma)$. Here, σ is the Gaussian kernel's scale and S is the list of all locations of people.

Problem formulation. We are given a set of labeled dataset of input-GT pairs ($\{x, y\} \in \mathcal{D}_{\mathcal{L}}$) and a set of unlabeled input data samples $x \in \mathcal{D}_{\mathcal{U}}$. The objective is to fit a mapping-function $f(x|\phi)$ (with parameters defined by ϕ) that accurately estimates target label y for unobserved samples. Note that this definition applies to both semi-supervised setting and synthetic-to-real transfer setting. In the case of synthetic-to-real transfer, the synthetic dataset is labeled and hence, can be used as the labeled dataset ($\mathcal{D}_{\mathcal{L}}$). Similarly, the real-world dataset is unlabeled and can be used as the unlabeled dataset ($\mathcal{D}_{\mathcal{U}}$).

In order to learn the parameters, both labeled and unlabeled datasets are exploited. Typically, loss functions such as L_1, L_2 or cross entropy error are used for labeled data. For exploiting unlabeled data $\mathcal{D}_{\mathcal{U}}$, existing approaches augment $f(x|\phi)$ with information like shape of the data manifold [208] via different techniques such as enforcing consistent regularization [209], virtual adversarial training [210] or pseudo-labeling [211]. In the proposed method, we employ pseudo-labeling based approach where we estimate pseudo-GT for unlabeled data, and then use them for supervising the network using traditional supervised loss functions.

Gaussian process. A Gaussian process (GP) $f(v)$ is an infinite collection of random variables, any finite subset of which have a joint Gaussian distribution. A GP is fully specified by its mean function ($m(v)$) and covariance function $K(v, v')$. These are defined below:

$$m(v) = \mathbb{E}[f(v)], \tag{10.1}$$

$$K\left(v, v'\right) = \mathbb{E}\left[\left(f(v) - m(v)\right)\left(f\left(v'\right) - m\left(v'\right)\right)\right], \tag{10.2}$$

where $v, v' \in \mathcal{V}$ denote the possible inputs that index the GP. The covariance matrix is computed from the covariance function K which expresses the notion of smoothness of the underlying function. GP can then be formulated as follows:

$$f(v) \sim \mathcal{GP}(m(v), K(v, v') + \sigma_\epsilon^2 I), \tag{10.3}$$

where I is identity matrix and σ_ϵ^2 is the variance of the additive noise. Any collection of function values is then jointly Gaussian as follows

$$f(V) = \left[f\left(v_1\right), \ldots, f\left(v_n\right)\right]^T \sim \mathcal{N}\left(\mu, K(V, V') + \sigma_\epsilon^2 I\right), \tag{10.4}$$

with mean vector and covariance matrix defined by the GP as mentioned earlier. To make predictions at unlabeled points, one can compute a Gaussian posterior distribution in closed form by conditioning on the observed data. For more details, we refer the reader to [212].

10.2 GP-based iterative learning

Figure 10.1 gives an overview of the proposed method. The network is constructed using an encoder $f_e(x, \phi_e)$ and a decoder $f_d(z, \phi_d)$, that are parameterized by ϕ_e and ϕ_d, respectively. The proposed framework is agnostic to the encoder network, and we show in the experiments section that it generalizes well to architectures such as VGG16 [140], ResNet-50 and ResNet-101 [213]. The decoder consists of a set of 2 conv-relu layers (see supplementary material for more details). Typically, an input crowd image x is forwarded through the encoder network to obtain the corresponding latent space vector z. This

vector is then forwarded through the decoder network to obtain the crowd density output y, *i.e.*,, $y = f_d(f_e(x, \phi_e), \phi_d)$.

We are given a training dataset, $\mathcal{D} = \mathcal{D}_\mathcal{L} \cup \mathcal{D}_\mathcal{U}$, where $\mathcal{D}_\mathcal{L} = \{x_l^i, y_l^i\}_{i=1}^{N_l}$ is a labeled dataset containing N_l training samples and $\mathcal{D}_\mathcal{U} = \{x_u^i\}_{i=1}^{N_u}$ is an unlabeled dataset containing N_u training samples. The proposed framework effectively leverages both the datasets by iterating the training process over labeled $\mathcal{D}_\mathcal{L}$ and unlabeled datasets $\mathcal{D}_\mathcal{U}$. More specifically, the training process consists of two stages: (i) Labeled training stage: In this stage, we employ supervised loss function $\mathcal{L}_s$ to learn the network parameters using labeled dataset, and (ii) Unlabeled training stage: We generate pseudo GTs for the unlabeled data points using the GP formulation, which is then used for supervising the network on the unlabeled dataset. In what follows, we describe these stages in detail.

10.2.1 Labeled stage

Since the labeled dataset $\mathcal{D}_\mathcal{L}$ comes with annotations, we employ L_2 error between the predictions and the GTs as supervision loss for training the network. This loss objective is defined as follows:

$$\mathcal{L}_s = \mathcal{L}_2 = \|y_l^{pred} - y_l\|_2, \tag{10.5}$$

where $y_l^{pred} = g(z_l, \phi_d)$ is the predicted output, y_l is the ground-truth, $z = h(x, \phi_e)$ is the intermediate latent space vector. Note that, the subscript l in the above quantities indicate that these are defined for labeled data.

Along with performing supervision on the labeled data, we additionally save feature vectors z_l^i's from the intermediate latent space in a matrix F_{z_l}.

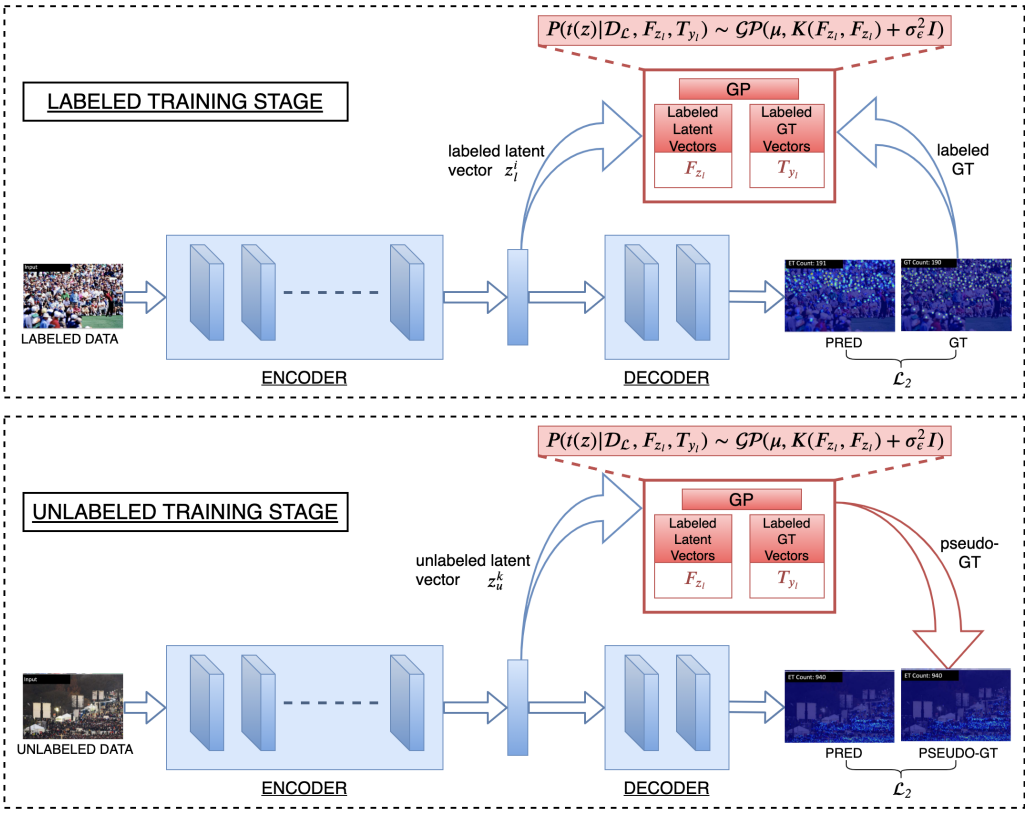


Figure 10.1: Illustration of the proposed framework. Training is performed iteratively over labeled and unlabeled data. For labeled data, we minimize the L_2 error between the predictions and GT. For unlabeled data, we minimize the L_2 error between the predictions and pseudo-GT.

Specifically, $F_{z_l} = \{z_l^i\}_{i=1}^{N_l}$. This matrix is used for computing the pseudo-GTs for unlabeled data at a later stage. The dimension of F_{z_l} matrix is $N_l \times M$. Here, M is the dimension of the latent space vector z_l. In our case, the latent space vector dimension is $64 \times 32 \times 32$ (see supplementary material for more details), which is reshaped to $1 \times 65,536$. Hence, $M = 65,536$.

10.2.2 Unlabeled stage

Since the unlabeled data $\mathcal{D}_{\mathcal{U}}$ does not come with any GT annotations, we estimate pseudo-GTs which are then used as supervision for training the network on unlabeled data. For this purpose, we model the relationship between the latent space vectors of the labeled images F_{z_l} along with the corresponding GT T_{y_l} and unlabeled latent space vectors z_u^{pred} jointly using GP.

Estimation of pseudo-GT: As discussed earlier, the training process iterates over labeled $\mathcal{D}_{\mathcal{L}}$ and unlabeled data $\mathcal{D}_{\mathcal{U}}$. After the labeled stage, the labeled latent space vectors F_{z_l} and their corresponding GT density maps T_{y_l} are used to model the function t which maps the relationship between the latent vectors and the output density maps as, $y = t(z)$. Using GP, we model this function $t(.)$ as an infinite collection of functions of which any finite subset is jointly Gaussian. More specifically, we jointly model the distribution of the function values $t(.)$ of the latent space vectors of the labeled and the unlabeled samples using GP as follows:

$$P(t(z)|\mathcal{D}_{\mathcal{L}}, F_{z_l}, T_{y_l}) \sim \mathcal{GP}(\mu, K(F_{z_l}, F_{z_l}) + \sigma_\epsilon^2 I), \tag{10.6}$$

where μ is the function value computed using GP, σ_ϵ^2 is set equal to 1, and K is the kernel function. Based on this, the conditional joint distribution for the latent space vector z_u^k of the k^{th} unlabeled sample x_u^k can be expressed as the following Gaussian distribution:

$$P(t(z_u^k)|\mathcal{D}_{\mathcal{L}}, F_{z_l}, T_{z_l}) = \mathcal{N}(\mu_u^k, \Sigma_u^k), \tag{10.7}$$

where

$$\mu_u^k = K(z_u^k, F_{z_l})[K(F_{z_l}, F_{z_l}) + \sigma_\epsilon^2 I]^{-1} T_{y_l}, \tag{10.8}$$

$$\Sigma_u^k = K(z_u^k, z_u^k) - K(z_u^k, F_{z_l})[K(F_{z_l}, F_{z_l}) + \sigma_\epsilon^2 I]^{-1} K(F_{z_l}, z_u^k) + \sigma_\epsilon^2 \tag{10.9}$$

where σ_ϵ^2 is set equal to 1 and K is a kernel function with the following definition:

$$K(Z, Z)_{k,i} = \kappa(z_u^k, z_l^i) = \frac{\langle z_u^k, z_l^i \rangle}{|z_u^k| \cdot |z_l^i|}. \tag{10.10}$$

Considering the large dimensionality of the latent space vector, $K(F_{z_l}, F_{z_l})$ can grow quickly in size especially if the number of labeled data samples N_l is high. In such cases, the computational and memory requirements become prohibitively high. Additionally, all the latent vectors may not be necessarily effective since these vectors correspond to different regions of images in terms of content and size/density of the crowd. In order to overcome these issues, we use only those labeled vectors that are similar to the unlabeled latent vector. Specifically, we consider only N_n nearest labeled vectors corresponding to an unlabeled vector. That is, we replace F_{z_l} by $F_{z_l,n}$ in Equation (10.7)-(10.9). Here $F_{z_l,n} = \{z_l^j : z_l^j \in nearest(z_u^k, F_{z_l}, N_n)\}$, and $T_{y_l,n} = \{y_l^j : z_l^j \in nearest(z_u^k, F_{z_l}, N_n)\}$ with $nearest(p, Q, N_n)$ being a function that finds top N_n nearest neighbors of p in Q.

The pseudo-GT for unlabeled data sample is given by the mean predicted in Equation (10.8), *i.e.*,, $y_{u,pseudo}^k = \mu_u^k$. The L_2 distance between the predictions $y_{u,pred}^k = g(z_u^k, \phi_e)$ and the pseudo-GT $y_{u,pseudo}^k$ is used as supervision for updating the parameters of the encoder $f_e(\cdot, \phi_e)$ and the decoder $f_d(., \phi_d)$.

Furthermore, the pseudo-GT estimated using Equation (10.8) may not be necessarily perfect. Errors in pseudo-GT will limit the performance of the network. To overcome this, we explicitly exploit the variance modeled by the

GP. Specifically, we minimize the predictive variance by considering Equation (10.9) in the loss function. As discussed earlier, using all the latent space vectors of labeled data may not be necessarily effective. Hence, we minimize the variance $\Sigma^k_{u,n}$ computed between z^k_u and the N_n nearest neighbors in the latent space vectors using GP. Thus, the loss function during the unlabeled stage is defined as:

$$\mathcal{L}_{un} = \frac{1}{|\Sigma^k_{u,n}|}\|y^k_{u,pred} - y^k_{u,pseudo}\|_2 + \log \Sigma^k_{u,n}, \tag{10.11}$$

where $y^k_{u,pred}$ is the crowd density map prediction obtained by forwarding an unlabeled input image x^k_u through the network, $y^k_{u,pseudo} = \mu^k_u$ is the pseudo-GT (see Equation (10.8)), and $\Sigma^k_{u,n}$ is the predictive variance obtained by replacing F_{z_l} in Equation (10.9) with $F_{z_l,n}$. Note that the prediction error (the first term) is scaled by loss by inverse of the variance. This ensures that the loss from uncertain pseudo-gts are down-weighted and hence, only accurate pseudo-gts are used for training.

10.2.3 Final objective function

We combine the supervised loss Equation (10.5) and unsupervised loss Equation (10.11) to obtain the final objective function as follows:

$$\mathcal{L}_f = \mathcal{L}_s + \lambda_{un}\mathcal{L}_{un}, \tag{10.12}$$

where λ_{un} is a hyper-parameter that weighs the unsupervised loss.

10.3 Experiments and results

In this section, we discuss the details of the various experiments conducted to demonstrate the effectiveness of the proposed method. Since the proposed

method is able to leverage unlabeled data to improve the overall performance, we performed evaluation in two settings: (i) *Semi-supervised settings*: In this setting, we varied the percentage of labeled samples from 5% to 75%. We first show that with the base network, there is performance drop due to the reduced data. Later, we show that the proposed method is able to recover a major percentage of the performance drop. (ii) *Synthetic-to-real transfer settings*: In this setting, the goal is to train on synthetic dataset (labeled), while adapting to real-world dataset. Unlabeled images from the real-world are available during training. In both settings, the proposed method is able to achieve better results as compared to recent methods.

10.3.1 Semi-supervised settings

In this section, we conduct experiments in the semi-supervised settings by reducing the amount of labeled data available during training. The rest of the samples in the dataset are considered as unlabeled samples wherever applicable. In the following sub-sections, we present comparison of the proposed method in the 5% setting with other recent methods. For comparison, we used 4 datasets: ShanghaiTech (SH-A/B)[1], UCF-QNRF [2], WorldExpo [59] and UCSD [53]. This is followed by a detailed ablation study involving different architectures and various percentages of labeled data used during training. For ablation, we chose ShanghaiTech-A and UCF-QNRF datasets since they contain a wide diversity of scenes and large variation in count and scales.

Implementation details. We train the network using Adam optimizer with a

Table 10.1: Comparison of results in SSL settings. Reducing labeled data to 5% results in performance drop by a big margin as compared to 100% data. ResNet-50 was used as the encoder network for all the methods. RL: Ranking-Loss. GP: Gaussian-Process. AG: Average Gain %[1].

Method	$\mathcal{D}_\mathcal{L}$	$\mathcal{D}_\mathcal{U}$	SH-A			SH-B			UCF-QNRF			WExpo		UCSD		
			MAE	MSE	AG	MAE	MSE	AG	MAE	MSE	AG	MAE	AG	MAE	MSE	AG
ResNet-50 (Oracle)	100%	-	76	126	-	8.4	14.5	-	114	195	-	10.1	-	1.7	2.1	-
ResNet-50 ($\mathcal{D}_\mathcal{L}$-only)	5%	-	118	211	-	21.2	34.2	-	186	295	-	14.2	-	2.2	2.8	-
ResNet-50+RL	5%	95%	115	208	2.0	20.1	32.9	4.0	182	291	1.7	14.0	0.01	2.2	2.8	0
ResNet-50+GP(Ours)	5%	95%	**102**	**172**	**16**	**15.7**	**27.9**	**22**	**160**	**275**	**10**	**12.8**	**10**	**2.0**	**2.4**	**12**

learning rate of $10e-5$ and a momentum of 0.9 on an NVIDIA Titan Xp GPU. We use batch size of 24. During training, random crops of size 256×256 are used. During inference, the entire image is forwarded through the network. We set aside 10% of the training set for the purpose of validation. The hyper-parameter λ_{un} was chosen based on the validation performance.

Comparison with recent approaches. Here, we compare the effectiveness of the proposed method with a recent method by Liu *et al.* [111] on 4 different datasets. In order to get a better understanding of the overall improvements, we also provide the results of the base network with (i) 100% labeled data supervision that is the oracle performance, and (ii) 5% labeled data supervision.

For all the methods (except oracle), we limited the labeled data used during training to 5% of the training dataset. Rest of the samples were used as unlabeled samples. We used ResNet-50 as the encoder network. The results of the experiments are shown in Table 10.1. We make the following observations for all the datasets: (i) Compared to using the entire dataset, reducing the labeled data during training (to 5%) leads to significant increase in error. (ii) The proposed GP-based framework is able to reduce the performance drop by a large margin. Further, the proposed method achieves an average gain

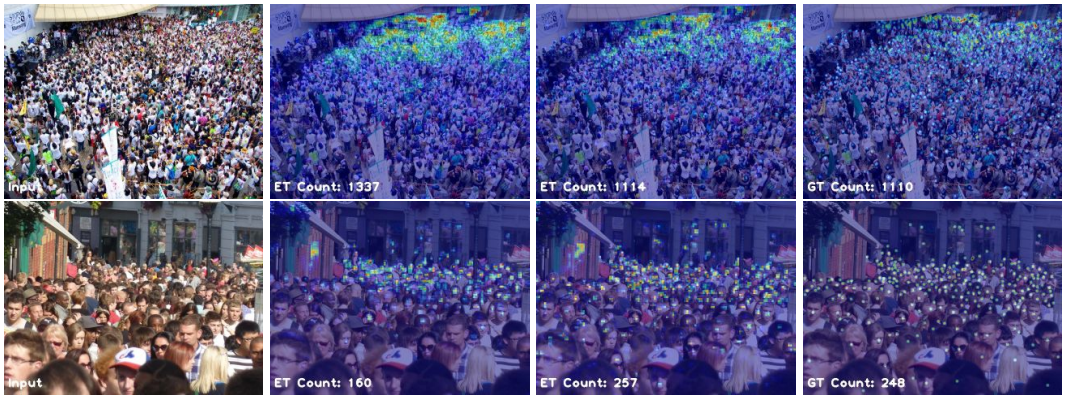
Table 10.2: Results of ablation study with different %-ages of labeled data. The proposed method achieves significant gains across different percentages of labeled data. We used ResNet-50 as the encoder network for all the experiments. AG: Average Gain %[1].

$\mathcal{D}_{\mathcal{L}}$ %	SH-A					UCF-QNRF				
	No-GP ($\mathcal{D}_{\mathcal{L}}$-only)		GP ($\mathcal{D}_{\mathcal{L}} + \mathcal{D}_{\mathcal{U}}$)		AG	No-GP ($\mathcal{D}_{\mathcal{L}}$-only)		GP ($\mathcal{D}_{\mathcal{L}} + \mathcal{D}_{\mathcal{U}}$)		AG
	MAE	MSE	MAE	MSE	%	MAE	MSE	MAE	MSE	%
5	118	211	102	172	16	186	295	160	275	10
25	110	160	91	149	12	178	252	147	226	14
50	102	149	89	148	6.1	158	250	136	218	13
75	93	146	88	139	4.7	139	240	129	210	9.8
100	76	126	-	-	-	114	195	-	-	-

(AG)[1] of anywhere between 10%-22% over the $\mathcal{D}_{\mathcal{L}}$-only baseline across all datasets. (iii) The proposed method is able to leverage the unlabeled data more effectively as compared to Liu *et al.* [111]. This is because the authors in [111] using a ranking loss on the unlabeled data which is based on the assumption that sub-image of a crowded scene is guaranteed to contain the same or fewer number of people compared to the entire image. We observed that this constraint is satisfied naturally for most of the unlabeled images, and hence it provides less supervision.

Ablation of labeled data percentage. We conducted an ablation study where we varied the percentage of labeled data used during the training process. More specifically, we used 4 different settings: 5%, 25%, 50% and 75%. The remaining data were used as unlabeled samples. We used ResNet-50 as the network encoder for all the settings. This ablation study was conducted on 2 datasets: ShanghaiTech-A (SH-A) and UCF-QNRF. The results of this ablation study are shown in Table 10.2. It can be observed for both datasets that as

[1] $AG = \frac{G_{mae}+G_{mse}}{2}$, $G_{mae} = \frac{mae_{(\mathcal{D}_{\mathcal{U}}+\mathcal{D}_{\mathcal{L}})} - mae_{(\mathcal{D}_{\mathcal{L}})}}{mae_{(\mathcal{D}_{\mathcal{L}})}}$, $G_{mse} = \frac{mse_{(\mathcal{D}_{\mathcal{U}}+\mathcal{D}_{\mathcal{L}})} - mse_{(\mathcal{D}_{\mathcal{L}})}}{mse_{(\mathcal{D}_{\mathcal{L}})}}$



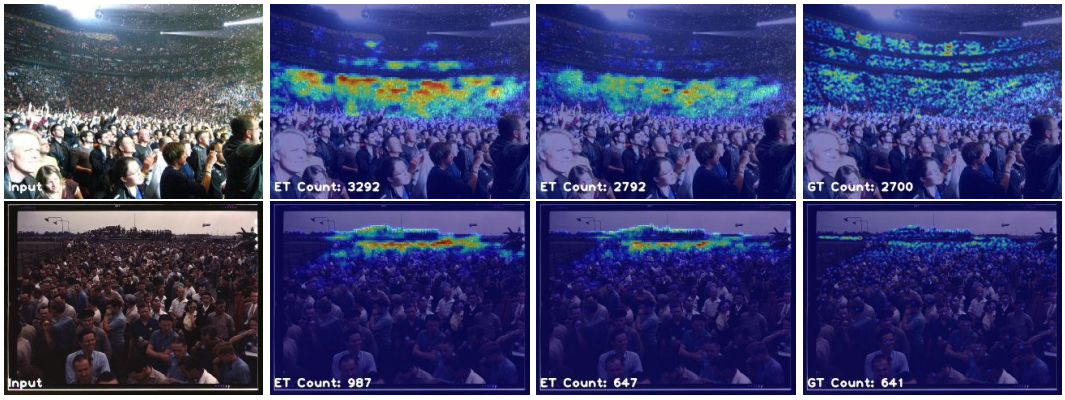
(a) (b) (c) (d)

Figure 10.2: Results of SSL experiments on the ShanghaiTech-A [1] dataset using the 5% labeled data setting. *(a):* Input. *(b)* No-GP *(c)* Proposed Method *(d)* Ground-truth.

Table 10.3: Results of ablation study with different networks. The proposed method is able to exploit unlabeled data irrespective of different architectures. We used 5% of the training data as labeled set, and the rest as unlabeled samples. AG: Average Gain %[1].

Net	$\mathcal{D}_\mathcal{L}$%	SH-A					UCF-QNRF				
		No-GP($\mathcal{D}_\mathcal{L}$-only)		GP($\mathcal{D}_\mathcal{L}+\mathcal{D}_\mathcal{U}$)		AG	No-GP ($\mathcal{D}_\mathcal{L}$-only)		GP ($\mathcal{D}_\mathcal{L}+\mathcal{D}_\mathcal{U}$)		AG
		MAE	MSE	MAE	MSE	%	MAE	MSE	MAE	MSE	%
ResNet-50	100	76	126	-	-	-	114	195	-		-
	5	118	211	102	172	16	186	295	160	275	10
ResNet-101	100	76	117	-	-	-	116	197	-	-	-
	5	131	200	110	162	18	196	324	174	288	11
VGG16	100	74	118	-	-	-	120	197	-		-
	5	121	205	112	163	14	188	316	175	291	7.4

the percentage of labeled data is reduced, the performance of the baseline network drops significantly. However, the proposed GP-based framework is able to leverage unlabeled data in all the cases to reduce this performance drop by a considerable margin. Figure 10.2 and 10.3 show sample qualitative results on ShanghaiTech-A and UCF-QNRF datasets for the semi-supervised protocol with 5% labeled data setting. It can be observed that the proposed method is able to predict the density maps more accurately as compared to the baseline method that does not consider unlabeled data.



(a) (b) (c) (d)

Figure 10.3: Results of SSL experiments on the UCF-QNRF [2] dataset using the 5% labeled data setting. *(a):* Input. *(b)* No-GP *(c)* Proposed Method *(d)* Ground-truth.

Architecture ablation. We conducted an ablation study where we evaluated the proposed method using different architectures. More specifically, we used different networks like ResNet-50, ResNet-101 and VGG16 as encoder network. The ablation was performed on 2 datasets: ShanghaiTech-A (SH-A) and UCF-QNRF. For all the experiments, we used 5% of the training dataset as labeled dataset, and the rest were used as unlabeled samples. The results of this experiment are shown in Table 10.3. Based on these results, we make the following observations: (i) Since networks like VGG16 and ResNet-101 have higher number of parameters, they tend to overfit more in the reduced-data setting as compared to ResNet-50. (ii) The proposed GP-based method obtains consistent gains by leveraging unlabeled dataset across different architectures.

Pseudo-GT Analysis. In order to gain a deeper understanding about the effectiveness of the proposed approach, we plot the histogram of normalized errors with respect to the predictions y^u_{pred} of the network and the pseudo-GT

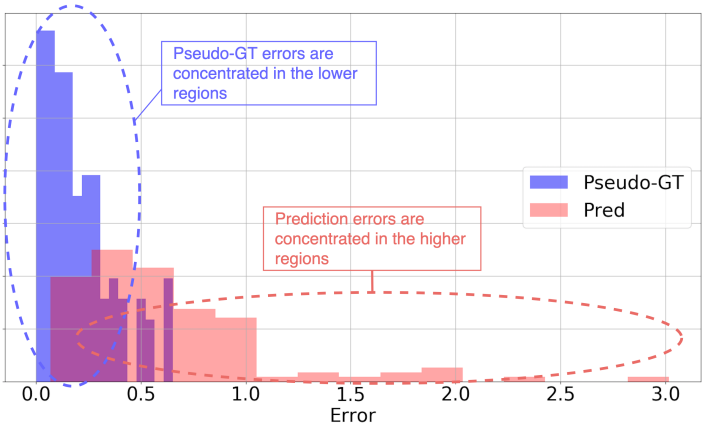


Figure 10.4: Histogram for pseudo-GT errors (err^u_{pseudo}) and prediction errors (err^u_{pred}) on unlabeled data during training. Note that pseudo-GT errors are concentrated on the lower end, implying that they are more closer to the ground truth as compared to the predictions. Hence, pseudo-GTs provide meaningful supervision.

y^u_{pseudo} for the unlabeled data during the training process. Specifically, we plot histograms of err^u_{pred} and err^u_{pseudo}, where $err^u_{pred} = \frac{|y^u_{pred} - y^u_{gt}|}{y^u_{gt}}$ and $err^u_{pseudo} = \frac{|y^u_{pseudo} - y^u_{gt}|}{y^u_{gt}}$. Here, y^u_{gt} is the actual GT corresponding to the unlabeled data sample. The plot is shown in Figure 10.4. It can be observed that the pseudo-GT errors are concentrated in the lower end of the error region as compared to the prediction errors. This implies that the pseudo-GTs are more closer to the GTs than the predictions. Hence, the pseudo-GTs obtained using the proposed method are able to provide good quality supervision on the unlabeled data.

10.3.2 Synthetic-to-Real transfer setting

Recently, Wang *et al.* [4] proposed a synthetic crowd counting dataset (GCC) that consists of 15,212 images with a total of 7,625,843 annotations. The primary purpose of this dataset is to reduce the annotation efforts by training the networks on the synthetic dataset, thereby eliminating the need for labeling.

Table 10.4: Comparison of results in synthetic-to-real transfer settings. We train the network on synthetic crowd counting dataset (GCC), and leverage the training set of real-world datasets without any labels. We used the same network as described in [4].

Method	SH-A		SH-B		UCF-QNRF		UCF-CC-50		WExpo
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
No Adapt	160	217	22.8	30.6	276	459	487	689	42.8
Cycle GAN [129]	143	204	24.4	39.7	257	401	405	548	32.4
SE Cycle GAN [4]	123	193	19.9	28.3	230	384	373	529	26.3
Proposed Method	**121**	**181**	**12.8**	**19.2**	**210**	**351**	**355**	**505**	**20.4**

However, due to a gap between the synthetic and real-world data distributions, the networks trained on synthetic dataset perform poorly on real-world images. In order to overcome this issue, the authors in [4] proposed a Cycle-GAN based domain adaptive approach that additionally enforces SSIM consistency. More specifically, they first learn to translate from synthetic crowd images to real-world images using SSIM-based Cycle-GAN. This transfers the style in the synthetic image to more real-world style. The translated synthetic images are then used to train a counting network (SFCN) that is based on ResNet-101 architecture.

While this approach improves the error over the baseline methods, its performance is essentially limited in the case of large distribution gap between real and synthetic images. Moreover, the authors in [4] perform a manual selection of synthetic samples for training the network. This selections ensures that only samples that are closer to the real-world images in terms of the count are used for training. Such a selection is not feasible in the case of unsupervised domain adaptation where we have no access to labels in the target dataset.

The proposed GP-based framework overcomes these drawbacks easily and can be extended to the synthetic-to-real transfer setting as well. We

(a) (b) (c) (d)

Figure 10.5: Results of Synthetic-to-Real transfer experiments on ShanghaiTech-A dataset. *(a):* Input. *(b)* No Adapt *(c)* Proposed Method *(d)* Ground-truth.

consider the synthetic data as labeled training set and real-world training set as unlabeled dataset, and train the network to leverage the unlabeled dataset. The results of this experiment are reported in Table 10.4. We used the same network (SFCN) and training process as described in [4]. As it can be observed, the proposed method achieves considerable improvements compared to the recent approach. Since we estimate the pseudo-GT for unlabeled real-world images and use it as supervision directly, the distribution gap that the network needs to handle is much lesser. This results in better performance compared to the domain adaptive approach [4]. Unlike [4], we train the network on the unlabeled data and hence, we do not need to perform any synthetic sample selection. Figure 10.5 and 10.6 show sample qualitative results on the ShanghaiTech-A and UCF-QNRF datasets for the synthetic-to-real transfer protocol. The proposed method is able to predict the density maps more accurately as compared to the baseline.

(a) (b) (c) (d)

Figure 10.6: Results of Synthetic-to-Real transfer experiments on the UCF-QNRF [2] dataset. *(a):* Input. *(b)* No Adapt. *(c)* Proposed Method. *(d)* Ground-truth.

10.4 Summary

We focused on learning to count in the crowd from limited labeled data. Specifically, we proposed a GP-based iterative learning framework that involves estimation of pseudo-GT for unlabeled data using Gaussian Processes, which is then used as supervision for training the network. Through various experiments, we show that the proposed method can be effectively used in a variety of scenarios that involve unlabeled data like learning with less data or synthetic to real-world transfer. In addition, we conducted detailed ablation studies to demonstrate that the proposed method generalizes well to different network architectures and is able to achieve consistent gains for different amounts of labeled data.

Chapter 11

JHU-CROWD++: Large-Scale Crowd Counting Dataset

We identify the next set of challenges that require attention from the crowd counting research community and collect a large-scale dataset collected under a variety of conditions. Existing efforts like UCF_CROWD_50 [6], World Expo '10 [59] and ShanghaiTech [214] have progressively increased the complexity of the datasets in terms of average count per image, image diversity *etc*. While these datasets have enabled rapid progress in the counting task, they suffer from shortcomings such as limited number of training samples, limited diversity in terms of environmental conditions, dataset bias in terms of positive samples, and limited set of annotations. Idrees *et al.* [2] proposed a new dataset called UCF-QNRF that alleviates some of these challenges. Most recently, Wang *et al.* [130] released a large-scale crowd counting dataset consisting of 5,109 images with 2.13 million annotations. Specifically, the images are collected under a variety of illumination conditions. Nevertheless, they do not specifically consider some of the challenges such as adverse environmental

conditions, dataset bias and limited annotation data[1].

To address these issues, we propose a new large-scale unconstrained dataset (JHU-CROWD++) with a total of 4,372 images (containing 1,515,005 head annotations) that are collected under a variety of conditions. Specific care is taken to include images captured under various weather-based degradations. Additionally, we include a set of distractor images that are similar to the crowd images that contain complex backgrounds which may be confused for crowd. Fig 11.1 illustrates representative samples of the images in the JHU-CROWD++ dataset under various categories. Furthermore, the dataset also provides a much richer set of annotations at both image-level and head-level. These annotations include point-wise annotations, approximate sizes, blur-level, occlusion-level, weather-labels, *etc*. We also benchmark several representative counting networks, providing an overview of the state-of-the-art performance.

(a) (b) (c) (d) (e)

Figure 11.1: Representative samples of the images in the JHU-CROWD++ dataset. (a) Overall (b) Rain (c) Snow (d) Haze (e) Distractors.

[1]Existing datasets provide only point-wise annotations.

Table 11.1: Comparison of different datasets. P: Point-wise annotations for head locations, O: Occlusion level per head, B: Blur level per head, S: Size indicator per head, $S^{\dagger}$: Approximate size (w×h), I: Image level labels.

Dataset	Num of Images	Num of Annotations	Avg Count	Max Count	Avg H×W	Weather degradations	Distractors	Type of annotations
UCSD [53]	2,000	49,885	25	46	158×238	✗	✗	P
Mall [19]	2,000	62,325	-	53	320×240	✗	✗	P
UCF_CROWD_50 [6]	50	63,974	1,279	4,543	2101×2888	✗	✗	P
WorldExpo '10 [59]	3,980	199,923	50	253	576×720	✗	✗	P
ShanghaiTech [1]	1,198	330,165	275	3,139	598×868	✗	✗	P
UCF-QNRF [2]	1,535	1,251,642	815	12,865	2,013×2,902	✗	✗	P
NWPU-CROWD [130]	5,109	2,133,238	418	20,033	2,311× 3,383	✗	✓	P
JHU-CROWD (ours)	4,250	1,114,785	262	7,286	900×1,450	✓	✓	P, O, B, S, I
JHU-CROWD++ (ours)	4,372	1,515,005	346	25,791	910×1,430	✓	✓	P, O, B, $S^{\dagger}$, I

11.0.1 Motivation and dataset details

As discussed earlier, existing datasets (such as UCF_CROWD_50 [6], World Expo '10 [59] and ShanghaiTech [214]) have enabled researchers to develop novel counting networks that are robust to several factors such as variations in scale, pose, view *etc*. Several recent methods have specifically addressed the large variations in scale by proposing different approaches such as multi-column networks [1], incorporating global and local context [147], scale aggregation network [107], *etc*. These methods are largely successful in addressing issues in the existing datasets, and there is pressing need to identify newer set of challenges that require attention from the crowd counting community.

In what follows, we describe the shortcomings of existing datasets and discuss the ways in which we overcome them:

(i) *Limited number of training samples*: Typically, crowd counting datasets have limited number of images available for training and testing. For example, ShanghaiTech dataset [1] has only 1,198 images and this low number of images results in lower diversity of the training samples. Due to this issue, networks trained on this dataset will have reduced generalization capabilities. Although

datasets like Mall [19], WorldExpo '10 [59] have higher number of images, it is important to note that these images are from a set of video sequences from surveillance cameras and hence, they have limited diversity in terms of background scenes and number of people. Most recently, Idrees *et al.* [2] addressed this issue by introducing a high-quality dataset (UCF-QNRF) that has images collected from various geographical locations under a variety of conditions and scenarios. Although it has a large set of diverse scenarios, the number of samples is still limited from the perspective of training deep neural networks.

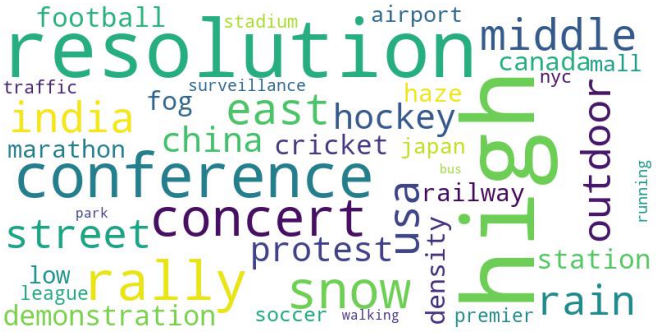


Figure 11.2: Summary of keywords used to scrape the internet for images.

To address this issue, we collect a new large scale unconstrained dataset with a total of 4,372 images that are collected under a variety of conditions. Such a large number of images results in increased diversity in terms of count, background regions, scenarios, *etc*. as compared to existing datasets. The images are collected from several sources on the internet using different keywords such as crowd, crowd+marathon, crowd+walking, crowd+India, *etc*. A summary of the keywords used for the search purpose is illustrated in Figure 11.2.

(a) (b) (c)

Figure 11.3: Examples of head-level annotations: (a) Dots (b) Approximate sizes (c) Blur-level.

(ii) *Absence of adverse conditions*: Typical application of crowd counting is video surveillance in outdoor scenarios which involve regular weather-based degradations such as haze, snow, rain *etc*. It is crucial that networks, deployed under such conditions, achieve more than satisfactory performance.

To overcome this issue, specific care is taken during our dataset collection efforts to include images captured under various weather-based degradations such as rain, haze, snow, *etc*. (as as shown in Figure 11.1(b-d)). Table 11.2 summarizes images collected under adverse conditions.

(iii) *Dataset bias*: Existing datasets focus on collecting only images with crowd,

Table 11.2: Summary of images collected under adverse conditions.

Degradation type	Rain	Snow	Fog/Haze	Total
No. of images	145	201	168	514
No. of annotations	40,328	47,347	48,821	136,496

due to which a deep network trained on such a dataset may end up learning bias in the dataset. Due to this error, the network will erroneously predict crowd even in scenes that do not contain crowd.

In order to address this, we include a set of distractor images that are similar to crowd images but contain very few people. These images can enable the network to avoid learning bias in the dataset. The total number of distractor images in the dataset is 106. Fig 11.1(e) shows sample distractor images.

(iv) *Limited annotations*: Typically, crowd counting datasets provide point-wise annotations for every head/person in the image, *i.e.,*each image is provided with a list of x, y locations of the head centers. While these annotations enable the networks to learn the counting task, absence of more information such as occlusion level, head sizes, blur level *etc*. limits the learning ability of the networks. For instance, due to the presence of large variations in perspective, size of the head is crucial to determine the precise count. One of the reasons for these missing annotations is that crowd images typically contain several people and it is highly labor intensive to obtain detailed annotations such as size.

To enable more effective learning, we collect a much richer set of annotations at both head-level/point-level and image-level. These are described below:

- Head-level/point-level annotations include x, y locations of heads and corresponding occlusion level, blur level and size level. The total number of point-level annotations in the dataset are 1,515,005. Occlusion label has three levels: {*un-occluded, partially occluded, fully occluded*}. Blur level has two labels: {*blur, no-blur*}. In JHU-CROWD [215], each head is labeled with a size indicator. We improve over these size annotations by providing "approximate" size (width and height) for each head annotation. To obtain these, annotators were instructed to annotate bounding boxes for a set of neighbouring heads which have similar sizes. *Note that these bounding boxes are only "approximate" and are not as accurate as the ones found in detection datasets*. Fig 11.3 illustrates sample annotations provided in our dataset.
- Image level annotations include scene-labels (such as *marathon, mall, railway station, stadium, etc.*) and the weather-labels (rain, snow and fog). Fig 11.4 illustrates the distribution of scene-labels in the proposed dataset.

11.0.2 Summary and evaluation protocol

Fig 11.1 illustrates representative samples of the images in the JHU-CROWD++ dataset under various categories. Table 11.1 summarizes the proposed dataset in comparison with the existing ones. It can be observed that the proposed dataset enjoys a host of properties such as a richer set of annotations, weather-based degradations and distractor images. With these properties, the proposed

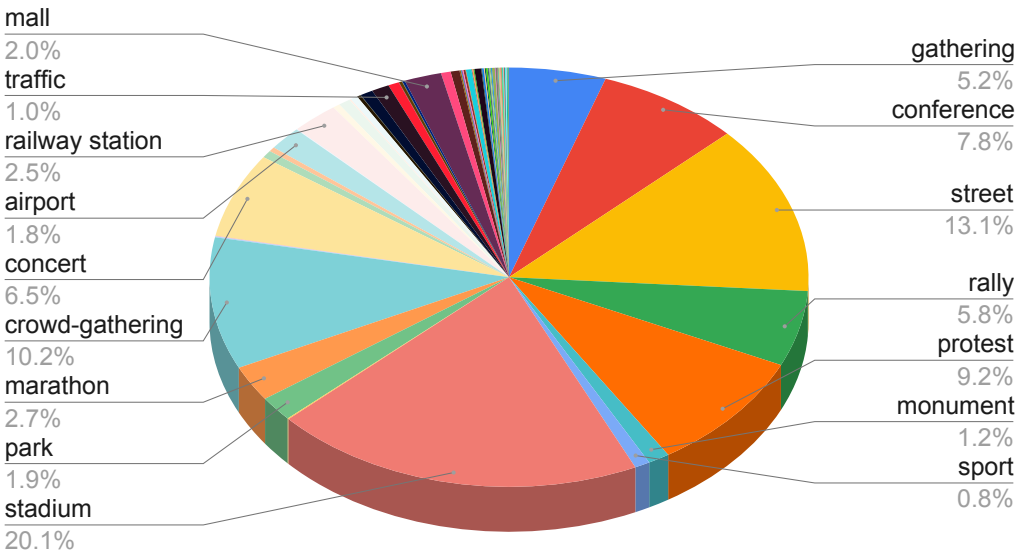


Figure 11.4: Distribution of image-level labels.

dataset will serve as a good complementary to other datasets such as UCF-QNRF and NWPU-CROWD. The dataset is randomly split into train, val and test sets, which contain 2722, 500 and 1600 images respectively.

Following the existing works, we perform evaluation using the standard MAE and MSE metrics. Furthermore, these metrics are calculated for the following sub-categories of images:

(i) Low density: images containing count between 0 and 50,

(ii) Medium density: images containing count between 51 and 500,

(iii) High density: images with count more than 500 people,

(iv) Weather degraded images, and

(v) Overall.

The metrics under these sub-categories will provide a holistic understanding

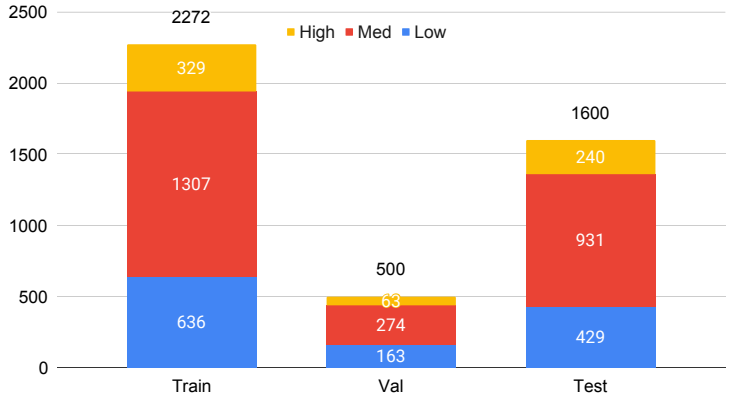


Figure 11.5: Distribution of images of different density levels in train, val and test sets.

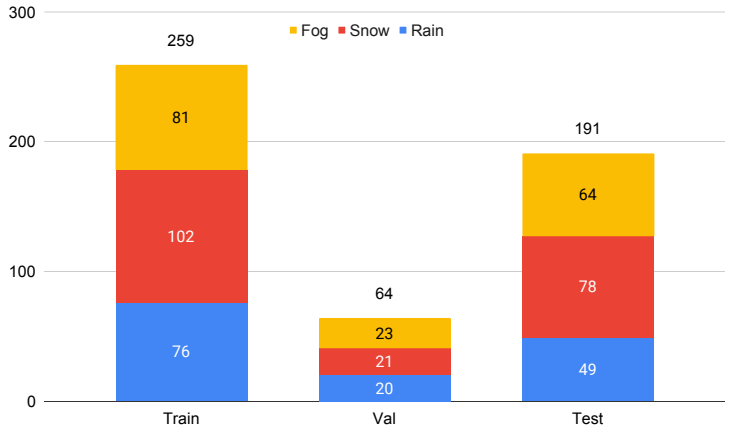


Figure 11.6: Distribution of images of weather conditions in train, val and test sets.

Table 11.3: Distribution of images under different densities.

Density	Low (0-50)	Med (51-500)	High (500+)	Total
No. of images	1,228	2,512	632	4,372

of the network performance.

Fig 11.5 and Fig 11.6 illustrate the distribution the number of images among the density and weather sub-categories respectively. Table 11.3 shows the distribution of images for different density-levels.

11.1 Benchmarking on JHU-CROWD++ dataset

In this section, we present results of benchmarking of several recent algorithms including the proposed method on the JHU-CROWD++ dataset. Specifically, we evaluate the following recent works: mulit-column network (MCNN) [1], cascaded multi-task learning for crowd counting (CMTL) [103], CSR-Net [161], SA-Net [107], context-aware crowd counting (CACC) [199], spatial fully convolutional network (SFCN) [4], deep structured scale integration network (DSSI-Net) [199], multi-level bottom-top and top-bottom feature fusion [118], Bayesian loss for counting (BCC) [125] and locate-size-count-CNN (LSC-CNN) [124] and CG-DRCN-CC [215].

All the networks are trained using the training set. We use the validation set for model selection. Table 11.4 and 11.5 show the results of the above experiments for various sub-categories of images. Based on these results we make the following observations:

(i) The proposed method (CG-DRCN-CC) with Res101 base network achieves lowest overall MAE while obtaining comparable performance for validation set.

(ii) The proposed method (CG-DRCN-CC) with Res101 base network achieves lowest overall MAE/MSE as compared to all the other methods on the test set. In addition, it achieves best errors for the “high-density" and “weather” categories while obtaining comparable performance for the rest of the categories.

(iii) The proposed method (CG-DRCN-CC) with VGG16 base network achieves comparable performance in all categories with respect to the other methods.

(iv) BCC [125] and LSC-CNN [124] achieve lowest errors in the “low-density"

Table 11.4: Results on JHU-CROWD++ dataset (**"Val Set"**). **RED** indicates best error and **BLUE** indicates second-best error.

Category	Low		Medium		High		Weather		Overall	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1] (CVPR 16)	90.6	202.9	125.3	259.5	494.9	856.0	241.1	532.2	160.6	377.7
CMTL [103] (AVSS 17)	50.2	129.2	88.1	170.7	583.1	986.5	165.0	312.9	138.1	379.5
CSR-Net [161](CVPR 18)	22.2	40.0	49.0	99.5	302.5	669.5	83.0	168.7	72.2	249.9
SA-Net [107](ECCV 18)	13.6	26.8	50.4	78.0	397.8	749.2	72.2	126.7	82.1	272.6
CACC [121] (CVPR 19)	34.2	69.5	65.6	115.3	336.4	**619.7**	101.8	179.3	89.5	**239.3**
SFCN [4] (CVPR 19)	11.8	19.8	39.3	73.4	297.3	679.4	**52.3**	**93.6**	62.9	247.5
DSSI-Net [199] (ICCV 19)	50.3	85.9	82.4	164.5	436.6	814.0	155.7	314.8	116.6	317.4
MBTTBF [118] (ICCV 19)	23.3	48.5	53.2	119.9	294.5	674.5	88.2	200.8	73.8	256.8
BCC [125](ICCV 19)	**6.9**	**10.3**	39.7	85.2	**279.8**	**620.4**	58.9	124.7	**59.3**	**229.2**
LSC-CNN [124] (PAMI 20)	**6.8**	**10.1**	**39.2**	**64.1**	504.7	860.0	77.6	187.2	87.3	309.0
CG-DRCN-CC-VGG16 (ours)	17.1	44.7	40.8	71.2	317.4	719.8	63.5	116.6	67.9	262.1
CG-DRCN-CC-Res101 (ours)	11.7	24.8	**35.2**	**57.5**	**273.9**	676.8	**54.0**	**106.8**	**57.6**	244.4

Table 11.5: Results on JHU-CROWD++ dataset (**"Test Set"**). **RED** indicates best error and **BLUE** indicates second-best error.

Category	Low		Medium		High		Weather		Overall	
Method	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1] (CVPR 16)	97.1	192.3	121.4	191.3	618.6	1,166.7	330.6	852.1	188.9	483.4
CMTL [103] (AVSS 17)	58.5	136.4	81.7	144.7	635.3	1,225.3	261.6	816.0	157.8	490.4
CSR-Net [161] (CVPR 18)	27.1	64.9	43.9	71.2	356.2	784.4	141.4	640.1	85.9	309.2
SA-Net [107] (ECCV 18)	17.3	37.9	46.8	69.1	397.9	817.7	154.2	685.7	91.1	320.4
CACC [121] (CVPR 19)	37.6	78.8	56.4	86.2	384.2	789.0	155.4	617.0	100.1	314.0
SFCN [4] (CVPR 19)	16.5	55.7	38.1	59.8	**341.8**	**758.8**	**122.8**	**606.3**	77.5	297.6
DSSI-Net [199] (ICCV 19)	53.6	112.8	70.3	108.6	525.5	1,047.4	229.1	760.3	133.5	416.5
MBTTBF [118] (ICCV 19)	19.2	58.8	41.6	66.0	352.2	760.4	138.7	631.6	81.8	**299.1**
BCC [125] (ICCV 19)	**10.1**	**32.7**	**34.2**	**54.5**	352.0	768.7	140.1	675.7	**75.0**	299.9
LSCCNN [124] (PAMI 20)	**10.6**	**31.8**	**34.9**	55.6	601.9	1,172.2	178.0	744.3	112.7	454.4
CG-DRCN-CC-VGG16 (ours)	19.5	58.7	38.4	62.7	367.3	837.5	138.6	654.0	82.3	328.0
CG-DRCN-CC-Res101 (ours)	14.0	42.8	35.0	**53.7**	**314.7**	**712.3**	**120.0**	**580.8**	**71.0**	**278.6**

categories. These methods do not follow the traditional density-estimation based approach for supervising the networks. Instead they incorporate size information during the training through strategies like Bayesian-loss and bounding box-based supervision.

(v) Res101-based methods tend to perform better compared to VGG16-based approaches in terms of overall error.

11.2 Summary

We introduced a new large scale unconstrained crowd counting dataset (JHU-CROWD++) consisting of 4,372 images with 1.51 million annotations. The new dataset is collected under a variety of conditions and includes images with weather-based degradations and other distractors. Additionally, the dataset provides a rich set of annotations such as head locations, blur-level, occlusion-level, approximate bounding boxes and other image-level labels. In addition, we benchmark several recent state-of-the-art crowd counting techniques on the new dataset.

Chapter 12

DAFE-FD: Density Aware Feature Enrichment for Face Detection

Face detection is an important step in many computer vision related tasks such as face alignment [216, 217], face tracking [218], expression analysis [219], recognition and verification [220], synthesis [221, 166]. Several challenges are encountered in face detection such as variations in pose, illumination, scale etc. Earlier CNN-based methods [83, 222, 223, 224], although mostly successful in handling variations in pose and illumination, performed poorly when detecting smaller faces. Recent methods [174, 10, 225, 226], based on CNN-based object detection frameworks such as Faster-RCNN or SSD, have focused particularly on smaller faces and have demonstrated promising results. In order to detect wide range of scales, these methods propose a two-pronged approach: (i) multi-scale detection and (ii) new anchor design strategies. In case of multi-scale detection, detectors are placed on different conv layers of the backbone network (VGG-16 [140] or ResNet [198]) to improve the discrepancies between object sizes and receptive fields.

Although this approach provided significant improvements over the earlier

(a) (b)

Figure 12.1: (a) Crowd density estimation on ShanghaiTech dataset using [1]. *Top row:* Input. *Middle row:* Ground truth density map. *Bottom row:* Estimated density map. (b) Face detection results using the proposed density enrichment module in the detection network.

single-scale methods, it is not capable of detecting extremely small sized faces (of the order 15×15). This stems from the fact that these methods are anchor-based approaches where detections are performed by classifying a pre-defined set of anchors generated by tiling a set of boxes with different scales and aspect rations on the image. While such approaches are relatively more robust in complicated scenes and provide computational advantages since inference time is independent of number of objects/faces (for single shot methods), their performance degrades significantly when used on smaller sized objects. The degradation is primarily due to a low overlap of ground truth boxes with the pre-defined anchor boxes and a mismatch between receptive fields of the feature maps and the smaller objects [10]. In order to overcome these drawbacks, recent methods have attempted to develop new anchor design strategies that involve intelligent selection of anchor scales and improved anchor matching strategy [174, 10].

While these recent methods address the drawbacks of anchor design or perform multi-scale detection, they do not emphasize on enhancing the feature maps for improving detection rates of small faces. To overcome this, we infuse information from crowd density maps to enrich the feature maps for addressing the problem of small face detection. Crowd density maps, originally used for counting in crowded scenarios, contain location information which can be exploited for improving detector performance. These density maps are especially helpful in the case of small faces, where traditional anchor-based classification loss may not be sufficient. Hence, we use density map based loss to provide additional supervision. Previous work [69, 227] have demonstrated considerable improvements by incorporating crowd density maps for applications like tracking. In this work, we propose to improve the feature maps by employing a density estimator module that performs the task of estimating the per pixel count of number of faces in the image. Figure 12.1(a) illustrates sample density estimation results using [1] along with the corresponding ground-truth. Figure 12.1(b) illustrates sample detection results by using the proposed density enrichment module into the detection network.

12.1 Proposed method

The proposed network architecture, shown in Figure 12.2, is a single stage detector based on VGG-16 architecture. The base network is built on Region Proposal Network (RPN) [198], which is a fully convolutional single stage network and takes an image of any size as input. However, unlike RPN that uses a single detector on conv5 layer, we use multiple detectors (D_1,D_2,D_3

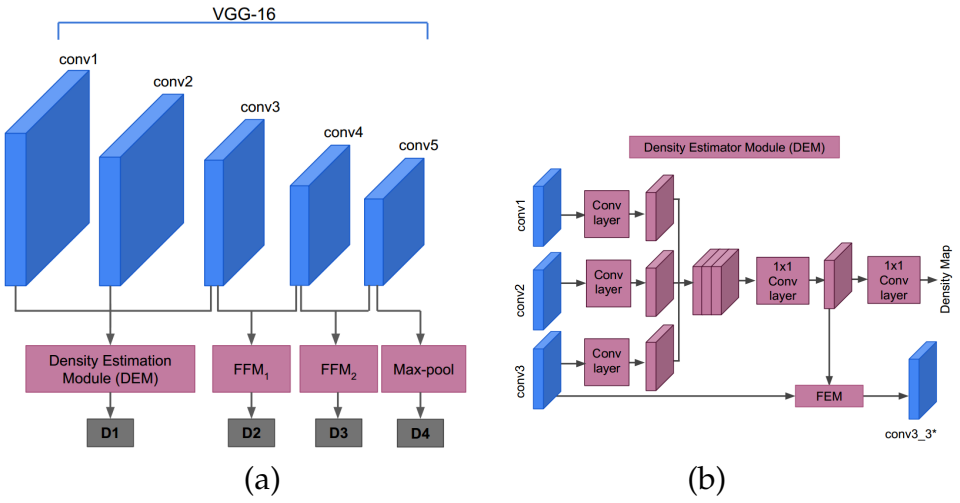


Figure 12.2: Overview. (a) Proposed network architecture: The network is based on VGG-16 and consists of 4 detectors D_1-D_4) to enable multi-scale detection. Feature maps (from conv3) for small face detector D_1 are enhanced by density estimator. FFM_1 and FFM_2 are feature fusion modules that are used to combine feature maps from different conv layers. (b) Density estimator module: Uses feature maps from first three conv layers of VGG-16 to estimate density map, which is further employed to enrich the conv3 feature maps for small face detection.

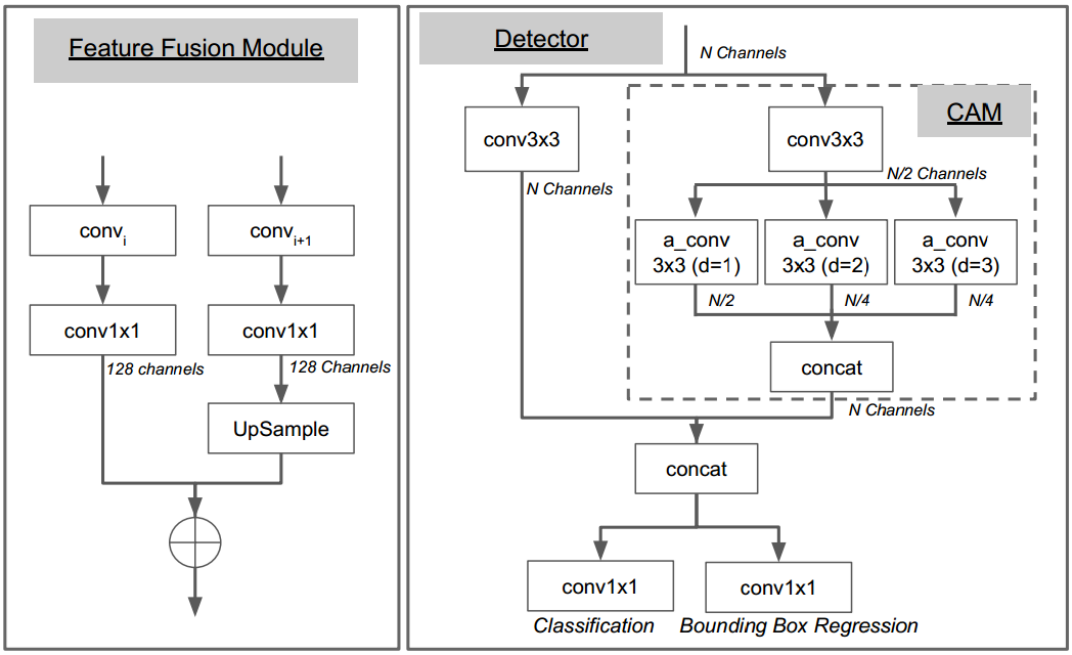
and D_4) on multiple conv layers [163]. These detectors, owing to the different receptive fields of the different conv layers, are better suited to handle various scales of objects, thereby improving the robustness of the network to different scales of faces present in the input image. However, in contrast to [163] that places the detectors on the conv layers of the base-network, we instead place the detectors on feature maps fused from multiple conv layers. In order to combine the feature maps, we employ a simple Feature Fusion Module (FFM) that effectively leverages semantic information present in different conv layers. Further, each detector consists of a Context Aggregation Module (CAM) followed by two sibling sub-networks: classification and a bounding box regression layer. The classification layer produces a score that represents the probability of finding a face defined by a specific anchor-box at a particular

location on the image (similar to [198]). The set of anchor boxes are formed similar to [198]. The bounding box regression layer computes the offsets with respect to the anchor boxes. These offsets are used to calculate the bounding-box co-ordinates of the predicted face.

Most importantly, the proposed network consists of a Density Estimator Module (DEM) that is the primary contribution of this work. This module predicts the density map associated with a particular input image and is incorporated into the detection network with the motivation of enriching the feature maps from conv layers before being used for small face detection. Recent methods [174, 10] employ new anchor design strategies to improve the detection of smaller faces and the feature maps are learned only through classification and bounding box regression loss, however, no specific emphasis is laid on the enhancement of feature maps. Considering this deficit, we propose to enrich the feature maps through an additional loss function from the density estimator module. This is also, partly, motivated by several earlier work [228, 229] that have employed multi-task learning to improve detection or classification performance. DEM is inspired by the success of recent CNN-based methods [59, 20, 147, 1, 104] for crowd counting which involve counting people in crowded images through density map regression. Furthermore, we propose a new fusion mechanism called Feature Enrichment Module to seamlessly combine the feature maps from conv layer of the base network with the output of DEM.

12.1.1 Feature Fusion Module (FFM)

Recent multi-scale object detection networks [162, 163] use multiple detectors on different conv layers. Although this technique provides considerable robustness to different scales, however, the detectors do not have access to feature maps from higher conv layers which have important semantic information. In order to leverage this high-level information, we employ a feature fusion module which takes input from i^{th} and $i+1^{th}$ conv layers and combines them as shown in Figure 12.3(a). First, the dimensionality of the feature maps of both conv layers is first reduced to 128 channels using 1×1 convolution. Since the dim-reduced feature maps from $i+1^{th}$ conv layer have lower resolution, they are upsampled using bilinear interpolation and then



(a) (b)

Figure 12.3: (a) Feature Fusion Module (b) Detector.

added to the dim-reduced feature maps from i^{th} conv layer. This is similar to [174], however, we extend this idea to add additional fusion modules to improve the performance. The proposed network has two fusion modules FFM_1 and FFM_2. FFM_1 fuses feature maps from conv3 and conv4, whereas FFM_2 fuses feature maps from conv4 and conv5.

Table 12.1: Anchor scales and feature strides for different detectors.

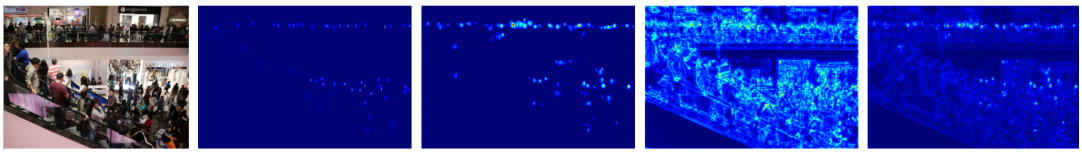
Detector	Input from	Stride	Anchor scales	Anchor sizes
D_1	DEM	4	1	16
D_2	FFM_1	8	1.5 2	24 32
D_3	FFM_2	16	4 8	64 128
D_4	conv5 max-pool	32	16 32	256 512

12.1.2 Multi-scale detectors

Multi-scale detection approaches [162, 163], that use multiple detectors on top of different conv layers, are known to introduce considerable robustness to scale variations and often perform as well as single scale detectors based on multi-image pyramid, thus providing additional advantage of computational efficiency. By adding detectors on earlier conv layers, these methods are able to match the receptive field sizes of the layers with objects of smaller sizes, thereby increasing the overlap between the anchor boxes and ground-truth boxes. Based on this idea, we add detectors D_1, D_2, D_3 and D_4. However, different from these earlier approaches that directly feed the output of conv layers to the detectors, we employ a different strategy as shown in Figure 12.2. D_1 receives features enriched by DEM, whereas D_2 and D_3 are placed on top of FFM_1 and FFM_2 respectively. D_4 is placed directly on top of max-pooled

version of conv5. The details of the feature strides and anchor scales are shown in Table 12.1. Each detector is constructed as shown in Figure 12.3(b).

Additionally, each detector is equipped with a Context Aggregation Module (shown in Figure 12.3 (b)) that integrates context information surrounding candidate bounding boxes. Context information has been used in several earlier work [230, 174] to improve the performance of detection systems. Zhu *et al.* [230] concatenated features pooled from larger windows and demonstrated significant improvement. Najibi *et al.* [174] used additional 5×5 and 7×7 convolutional filters to increase the receptive field size, in a way, imitating the strategy of pooling features from larger windows. While they achieved appreciable improvements, the use of large filter sizes results in more computations. Hence, we replace these large filters with atrous convolutions of size 3×3 [231, 232, 231] and different dilation factors. With the help of atrous convolutions, we are able to enlarge the receptive field size with minimal increase in computations.



(a) (b) (c) (d) (e)

Figure 12.4: Feature enrichment using density maps. (a) Input (b) Ground truth density (c) Estimated density (d) conv3 features before enhancement (e) conv3 features after enhancement.

12.1.3 Density Estimator Module

Recent crowd counting methods [59, 20, 147, 1, 104], that employ CNN-based density estimation techniques, have demonstrated promising results in complex scenarios. These techniques perform the task of counting people by estimating the density maps which represent the per pixel count of people in the image (as shown in Figure 12.1). For training, the ground-truth density map (D)for an input image is calculated using $D(x) = \sum_{x_g \in S} \mathcal{N}(x - x_g, \sigma)$, where σ is scale parameter of 2D Gaussian kernel and S is the set of all points at which people are located. Most crowd counting datasets provide 2d location of people in the input images as annotations. Figure 12.1 illustrates a few sample input and ground-truth density map pairs along with corresponding density map estimated using a recent technique [147]. It can be observed that, in spite of heavy occlusions and presence of extremely small scales, these recent techniques are able to estimate high quality density maps and count with reasonably low error.

While the success of these methods is attributed mostly to the use of advanced CNN architectures, reformulating the problem of counting as a density map regression also played an important role in their success. As compared to the earlier detection-based counting approaches [233, 234], these recent methods are able to achieve success due to the reformulation. By reformulating, these methods are able to avoid the problems of occlusion and tiny scales by letting the network take care of such variations. Hence, we explore the use of density estimation to incorporate robustness towards occlusion and tiny scales in the face detection network. In part, this contribution is also

inspired by recent methods [228, 229, 180] that learn multiple related tasks using multi-task learning. These methods have demonstrated considerable gains in performance when they train their network to perform additional auxiliary tasks.

To incorporate the task of density estimation in the detection network, we include a density estimator module. Recent crowd counting and density estimation approaches [1, 104, 147, 61] are based on multi-scale and multi-column networks, where the input image is processed by different CNN columns with varied receptive field sizes. The use of different columns results in increased robustness towards scale variations. Motivated by these approaches, we construct the density estimator module as shown in Figure 12.2(b). Instead of processing the input images through different networks as in [1], we use feature maps from the base network, thereby minimizing the computations. Our strategy is to mimic the multi column networks structures [1] by considering feature maps conv1, conv2 and conv3 layers of VGG-16, which correspond to different receptive field sizes. DEM first downsamples the feature maps from conv1 and conv2 layers using max-pooling to match the size of feature maps from conv3 layer. After resampling, the dimensionality of the feature maps is reduced to minimize computations and memory requirement, followed by additional convolutions and concatenation. The concatenated feature maps are processed by 1×1 conv layer to produce the final density map. Following loss function is used to obtain the network weights: $L_{den} = \frac{1}{N}\sum_{i=1}^{N} \|F_d(X_i, \Theta) - D_i\|_2$, where, N is number of training samples, X_i is the i^{th} input image, $F_d(X_i, \Theta)$ is the estimated density, D_i is the

i^{th} ground-truth density and Θ corresponds to network weights.

Feature Enrichment Module. We use the output of DEM to enhance the feature maps from conv3 layer in order to improve detection rates of smaller faces. Since the detector on conv3 has the smallest scale and is responsible for detecting the smaller faces, we choose to fuse information from DEM into conv3 feature maps. Various fusion techniques, such as feature concatenation or multiplication or addition, are available to incorporate information from DEM into the face detector network. However, these methods are not necessarily effective. Since the feature maps produced by DEM are used for density estimation, they have largely different range as compared to feature maps corresponding to conv layers from the detection network and hence, they cannot be directly fused with feature maps from conv3 layer through simple techniques such as addition or concatenation. As pointed out in [235], this problem is commonly encountered in networks that attempt to combine feature maps from different conv layers [236]. Liu *et al.* [235] introduce a L2-normalization based scaling technique to overcome this problem. Although this method is successfully used in different works [230], it did not perform promisingly in our case for the following reasons. First, the range of the feature maps from DEM is vastly different from that of conv3 feature maps and this gap is significantly wider as compared to other problems [230] where [235] has worked successfully. Second, the intermediate feature maps from the DEM have significantly low number of channels and hence, their dimensionality needs to increased to match that of feature maps from conv3 layer in order to perform a addition or multiplication based fusion.

Based on these considerations, we propose a simple Feature Enrichment Module (FEM) that avoids the challenges discussed above. Instead of using intermediate feature maps from DEM, we directly employ its density map output. The feature maps (f_3) from conv3 of the base-network are modified using the estimated density map as follows: $f_3 = f_3 + \alpha f'_d$, where, α is a learnable scaling factor and f'_d is f_d replicated 256 times to match the dimensionality of conv3 feature maps. Figure 12.4 illustrates feature maps from conv3 layer before and after enrichment. It can be easily observed from this figure that the features at the location of small faces get enhanced while those at other locations get suppressed.

12.1.4 Loss function

The weights of the proposed network are learned my minimizing the following multi-task loss function: $L = L_{cls} + \lambda_b L_{box} + \lambda_d L_{den}$, where, L_{cls} is face classification loss, L_{box} is bounding-box regression loss and L_{den} is density estimation loss. L_{cls} and L_{box} are defined as follows:

$$L_{cls} = \sum_{m=1}^{4} \frac{1}{N_m^c} \sum_{i \epsilon A_m} l_{ce}(p_i, p'_i) \tag{12.1}$$

$$L_{box} = \sum_{m=1}^{4} \frac{1}{N_m^r} \sum_{i \epsilon A_m} p_i l_{reg}(t_i, t'_i), \tag{12.2}$$

where, l_{ce} is standard cross entropy error, m indexes over the four detectors D_1-D_4, A_m are the set of anchors in detector D_m, p_i and p'_i are ground-truth and predicted labels respectively for the i^{th} anchor box, N_m^c is the number of anchors selected in the detector D_m and is used to normalize the classification loss, l_{reg} is bounding box regression loss for each positively labelled anchor

box. Similar to [198], the regression space is parametrized with a log-space shift and a scale invariant translation. Smooth l_1 loss is used as l_{reg}. In this new space, t_i is the regression target and t'_i is predicted co-ordinates. N^r_m is the number of positively labelled anchor boxes that are selected for the computing the loss and is used to normalize the bounding box loss. λ_b and λ_d are scaling factors to balance the loss function.

12.1.5 Training

Training details. The network is trained on a single GPU using stochastic gradient descent (momentum = 0.9 and weight decay = 0.0005) for $120k$ iterations. The learning rate is initially set to 0.001 and is dropped by a factor of 10 at $100k$ and $115k$ iterations. Anchor boxes are generated using the scales shown in Table 12.1 with a base anchor size of 16 pixels. Anchor boxes are labelled positively if their overlap (intersection over union) with ground truth boxes is greater than 0.5 and are negatively labelled if the overlap is below 0.3. A total of 256 anchor boxes per detector are selected for each image to compute the loss. The selection is performed using online hard example mining (OHEM) technique [158], where negatively labelled anchors with highest scores and positively labelled with lowest scores are selected. Such a selection procedure results in faster and stable training as compared to random selection [158]. The ground-truth density maps for training DEM are obtained using the method described in Section 12.1.3. The face annotations provided by the datasets are used to compute the points where faces are located and hence, no extra annotations are required. For inference, 1000 best scoring anchors from each

detector are selected as detections, followed by a non maximal suppression (NMS) with a threshold of 0.3.

Dataset details. The network is trained using WIDER dataset [5] which consists of 32,203 images with 393,703 annotated faces. The dataset presents a variety of challenges such as wide variations in scale and difficult occlusions. It is divided into training, validation and test set using a 40:10:50 ratio. For evaluation purpose, the dataset has been further divided into three categories: Easy, Medium and Hard. The detector performance is measured using mean average precision (mAP) with a intersection over union (IoU) threshold of 0.5.

12.2 Experiments and Results

In this section, we discuss details of the experiments and results on different datasets. Additionally, we present the results of an ablative study on WIDER validation set to explain the effect of different modules present in the proposed network.

12.2.1 WIDER

As discussed earlier, WIDER dataset consists of validation and test splits. We use the validation set to perform an ablative study to explain the effects of different modules in the proposed network. For this study, we use a single scale of the input image (no multi-image pyramid) similar to [174]. In addition, comparison of results on validation and test set with recent methods is presented.

Table 12.2: Ablation study Results (AP) on WIDER [5] validation.

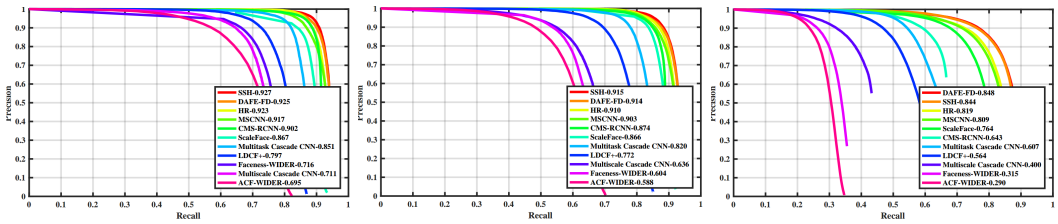
Category	Mehthod	Easy	Medium	Hard
Baseline	Baseline	91.0	89.9	80.6
Context	Baseline + Context [174]	91.6	90.2	81.8
	Baseline + CAM	91.9	90.6	82.4
Density estimator	Baseline + CAM + DEM (add)	92.0	90.6	82.3
	Baseline + CAM + DEM (concat)	92.0	90.6	82.4
	Baseline + CAM + DEM (FEM) ($\lambda_d = 0$)	92.1	90.6	82.5
	Baseline + CAM + DEM (FEM) ($\lambda_d = 1$)	92.4	90.8	83.2

Ablation study. To understand the effects of different modules in the proposed network, we experimented with 3 broad configurations as shown in Table 12.2. The results of these configurations are analyzed below:

(i) Baseline: This configuration uses VGG-16 as the base-network along with feature fusion module and 4 detectors D_1-D_4. Results of this network is considered as baseline performance and through addition of different modules, we demonstrate the improvements with respect to this baseline.

(ii) Baseline with context: Earlier work [174, 230] have already demonstrated the importance of incorporating context in the detection network. Similar observations are made in our experiments. By using a context processing module similar to [174], an improvement of 1.2% in the mean average precision (mAP) score for hard faces is obtained. Further, the use of atrous based context aggregation increased the mAP score by another 0.6% resulting in an overall improvement of 1.8%.

(iii) Baseline with context and DEM. In this case, we analyze the effect of incorporating DEM into the detection network. First, we experimented with different ways of integrating the feature maps from DEM into detection network through feature addition and concatenation, where the feature maps



(a) Easy (b) Medium (c) Hard

Figure 12.5: Precision-recall curves on WIDER test dataset[5]

from the penultimate layer of DEM are expanded through 1×1 convolutions to match the dimensionality of conv3 feature maps, followed by addition/multiplication of these two feature maps. It can be observed from Table 12.2, that these two configurations do not result in any improvement of the mAP scores. This is primarily due to vast difference in the scales of the feature maps (as discussed in Section 12.1.3).

Figure 12.6: Detection results of the proposed method on WIDER dataset[5].

Next, we added the feature enrichment module (FEM) to enhance the conv3 feature maps. This resulted in an overall improvement of 0.8% in mAP score for hard faces as compared to the baseline with context (CAM), thus demonstrating the significance of the proposed feature enrichment module and density estimator. Furthermore, in order to ensure that the improvements obtained are due to density estimation loss, we conducted another experiment with $\lambda_d = 0$ and no changes with respect to the baseline with CAM

configuration was observed.

Table 12.3: Comparison of results (AP) on WIDER [5] validation.

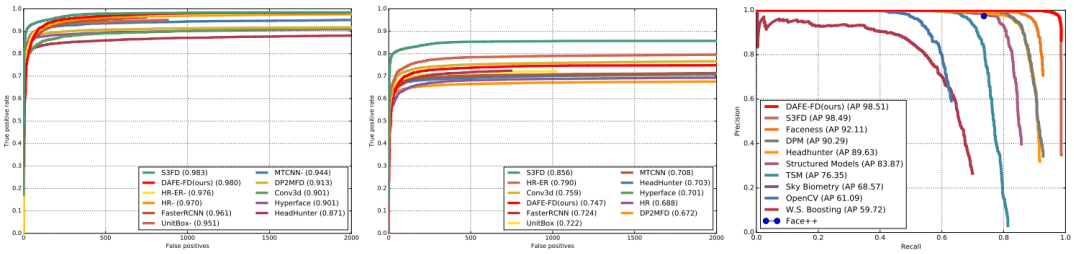
Method	Easy	Medium	Hard
CMS-RCNN [230]	89.9	87.4	62.9
HR-VGG16 + Pyramid [9]	86.2	84.4	74.9
HR-ResNet101 + Pyramid [9]	92.5	91.0	80.6
SSH [174]	91.9	90.7	81.4
SSH + Pyramid [174]	93.1	92.1	84.5
Face-MagNet [237]	92.0	91.3	85.0
S3FD + Pyramid [10]	93.7	92.4	85.2
DAFE-FD (ours)	92.4	90.8	83.2
DAFE-FD + Pyramid (ours)	93.4	92.2	85.2

Table 12.4: Comparison of results (AP) on WIDER [5] test.

Method	Easy	Medium	Hard
LDCF+[238]	79.7	77.2	56.4
MT-CNN [239]	85.1	82.0	60.7
CMS-RCNN [230]	89.9	87.4	62.9
HR-VGG16 + Pyramid [9]	86.2	84.4	74.9
HR-ResNet101 + Pyramid [9]	92.5	91.0	80.6
SSH + Pyramid [174]	92.7	91.5	84.4
Face-MagNet [237]	91.2	90.5	84.4
S3FD + Pyramid [10]	92.8	91.3	84.0
DAFE-FD + Pyramid (ours)	92.5	91.4	84.8

Comparison with other methods. We compare the results of the proposed method with recent state-of-the-art methods such as SSH [174], Face-MagNet [237], S3FD [10], HR [9], CMS-RCNN [230], MT-CNN [239], LDCF [238], Faceness [240] and Multiscale Cascaded CNN [5]. For the validation set, the results of the proposed method are obtained using single-scale inference as well as image-pyramid based reference (as shown in Table 12.3). It can be observed that DAFE-FD using single-scale inference achieves superior results as compared to HR that is based on image pyramid. Furthermore,

DAFE-FD (single-scale) achieves better results as compared to SSH-single-scale (recent best method) in all the subsets of WIDER dataset. Specifically, an improvement of 1.8% in case of “hard" set is obtained. Further improvements are attained by using pyramid-based inference and the proposed method is able to outperform SSH-pyramid and achieve comparable results with respect to S3FD. It is important to note that S3FD is based on single-shot detection approach and it involves extra detectors and feature maps from conv6 and conv7 layers in addition to the use of data augmentation based on multi-scale cropping and photometric distortion [241]. In spite of these additional factors in case of S3FD, DAFE-FD achieves comparable performance with respect to S3FD on the validation set, while obtaining better results on the test set as described below.



(a) (b) (c)

Figure 12.7: Comparison of results on different datasets (a) FDDB discrete score [7] (b) FDDB continuous score [7](c) Pascal faces Pascal-faces [8]. Note that HR/HR-ER [9] uses FDDB for training and evaluate using 10-fold cross-validation. S3FD [10] and Conv3D [11] generate ellipses to reduce localization error. Moreover, in case of S3FD, the authors manually annotate many unlabelled faces in FDDB dataset that results in improved performance. In contrast to these methods, we use FDDB and Pascal faces for testing only and employ rectangular bounding box to evaluate the results.

The average precision scores of the proposed method on the test set of WIDER dataset are shown in Table 4 and the corresponding precision-recall curves are shown in Figure 12.5. It can be clearly observed that DAFE-FD

outperforms existing state-of-the-art methods on the “hard" subset while achieving comparable or better performance on the other subsets. Detection results are shown in Figure 12.6.

12.2.2 FDDB

This dataset consists of 2,845 images with a total of 5,171 annotated faces. Figure 12.7 (a) and (b) shows comparison of ROC curves for different methods (S3FD[10], HR/HR-ER [9], Faster RCNN, UnitBox [242], MT-CNN [239], D2MFD [243], Conv3D [11], Hyperface [180] and Headhunter [224]) with the proposed method in discrete and continuous mode respectively. We use rectangular bounding boxes for evaluation as opposed to HR, S3FD and Conv3D that use elliptical regression to reduce localization error. Also, in contrast to HR that is trained on FDDB, we do not use images in FDDB for training purpose. In spite of lacking these additional features, DAFE-FD achieves consistently better performance in case of discrete scores and is comparable to other methods in case of continuous scores. Although S3FD obtains slightly better performance, it is important to consider that the authors manually annotated several unlabelled faces in the FDDB dataset which results in increased performance.

12.2.3 Pascal Faces

This dataset [8] consists of 851 images with a total of 1,355 labelled faces and it is a subset of the PASCAL person layout dataset [244]. Figure 12.7(c) shows the comparison of precision-recall curves on this dataset for different methods

with the proposed method. The proposed DAFE-FD method outperforms existing methods such as S3FD [10], Faceness [240], DPM [223], Headhunter [224] and many others.

12.3 Summary

We proposed a feature enrichment technique to improve the performance of small face detection. In contrast to existing methods that employ new strategies to improve anchor design, we instead focus on enriching the feature maps directly which is inspired by crowd counting/density estimation techniques that estimate the per pixel density of people/faces present in an image. Experiments conducted on different datasets, such as WIDER, Pascal-faces and FDDB, demonstrate considerable gains in performance due to the use of proposed density enrichment module. Additionally, the proposed method is complementary to recent improvements in anchor designs and hence, it can be used to obtain further improvements.

Chapter 13

Conclusions and Future Work

Crowd counting is an important problem owing to its usage in a wide range of applications such as video surveillance, traffic monitoring, public safety and urban planning, cell microscopy, agricultural and environmental monitoring. Considering its wide impact across a range of domains, we focus on developing Single Image based Crowd Counting Approaches using Deep Learning Techniques. Through these approaches, we explicitly attempt to address a variety of challenges encountered by the crowd counting research community. Specifically, we tackle the problem of large variations in scales and appearances of objects by incorporating context information at global and local levels. In order to address the issue of background clutter, we exploit different attention mechanisms like inverse attention and hierarchical attention which result in significant improvements over existing approaches. Additionally, we design fusion strategies such as multi-level and multi-path schemes and uncertainty-based iterative residual aggregation which can effectively combine information from multiple layers in a deep network for producing scale robust models.

In addition to the task related challenges, we also address problems related to the data. Specifically, we develop techniques to train deep networks from limited labeled data while exploiting large amounts of unlabeled or weakly-labeled samples. In this attempt, we propose Gaussian Processes based semi-supervised learning technique and class activation map based pseudo-label generation training approach for the weakly labeled data. Furthermore, we also introduce a new large-scale crowd counting dataset which can be used to train considerably larger networks. The proposed data consists of 4,372 high resolution images with 1.51 million annotations. We made explicit efforts to ensure that the images are collected under a variety of diverse scenarios and environmental conditions. The dataset provides a richer set of annotations like dots, approximate bounding boxes, blur levels, etc.

13.1 Future Research Directions

While this thesis presents significant progress in single image based crowd counting, more work is required in the following areas.

- There is a need to further improve the counting errors in order to enable them for critical real world applications.
- Existing crowd counting approaches suffer from poor cross-dataset performance due to issues such as domain/distributional shift. Hence, there is a pressing need to develop domain adaptation/transfer learning approaches to ensure better generalization abilities.
- Existing crowd counting approaches are overly focused on the learning

to estimate the count from single images and video-based counting has received less importance. Considering that video data is sequential and provides more information as compared to single images, it would be beneficial to exploit this modality for achieving better performance improvements.

Bibliography

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597.

[2] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *European Conference on Computer Vision*. Springer, 2018, pp. 544–559.

[3] H. Idrees, K. Soomro, and M. Shah, "Detecting humans in dense crowds using locally-consistent scale prior and global occlusion reasoning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 10, pp. 1986–1998, 2015.

[4] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," *arXiv preprint arXiv:1903.03303*, 2019.

[5] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.

[6] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554.

[7] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009*, vol. 2, no. 7, p. 8, 2010.

[8] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, no. 10, pp. 790–799, 2014.

[9] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 951–959.

[10] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3fd: Single shot scale-invariant face detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] Y. Li, B. Sun, T. Wu, and Y. Wang, "Face detection with end-to-end integration of a convnet and a 3d model," in *European Conference on Computer Vision*. Springer, 2016, pp. 420–436.

[12] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded scene analysis: A survey," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 3, pp. 367–386, 2015.

[13] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 345–357, 2008.

[14] D. Kang, Z. Ma, and A. B. Chan, "Beyond counting: Comparisons of density maps for crowd analysis tasks-counting, detection, and tracking," *arXiv preprint arXiv:1705.10118*, 2017.

[15] F. Xiong, X. Shi, and D.-Y. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *IEEE International Conference on Computer Vision*. IEEE, 2017.

[16] Y. Wang and Y. Zou, "Fast visual object counting via example-based density estimation," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3653–3657.

[17] E. Walach and L. Wolf, "Learning to count with cnn boosting," in *European Conference on Computer Vision*. Springer, 2016, pp. 660–676.

[18] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332.

[19] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting." in *European Conference on Computer Vision*, 2012.

[20] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 615–629.

[21] S. Zhang, G. Wu, J. P. Costeira, and J. M. F. Moura, "Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *IEEE International Conference on Computer Vision*. IEEE, 2017.

[22] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Understanding traffic density from large-scale web camera data," in *IEEE Computer Vision and Pattern Recognition*. IEEE, 2017.

[23] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal networks," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[24] E. Toropov, L. Gui, S. Zhang, S. Kottur, and J. M. Moura, "Traffic flow from a low frame rate city camera," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3802–3806.

[25] G. French, M. Fisher, M. Mackiewicz, and C. Needle, "Convolutional neural networks for counting fish in fisheries surveillance video," in *British Machine Vision Conference Workshop*. BMVA Press, 2015.

[26] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PloS one*, vol. 5, no. 4, p. e10047, 2010.

[27] H. Blumer, "Collective behavior," *New outline of the principles of sociology*, pp. 166–222, 1951.

[28] A. F. Aveni, "The not-so-lonely crowd: Friendship groups in collective behavior," *Sociometry*, pp. 96–99, 1977.

[29] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Reviews of modern physics*, vol. 81, no. 2, p. 591, 2009.

[30] L. F. Henderson, "The Statistics of Crowd Fluids," vol. 229, pp. 381–383, Feb. 1971.

[31] J. K. Parrish and L. Edelstein-Keshet, "Complexity, pattern, and evolutionary trade-offs in animal aggregation," *Science*, vol. 284, no. 5411, pp. 99–101, 1999.

[32] H.-P. Zhang, A. Be'er, E.-L. Florin, and H. L. Swinney, "Collective motion and density fluctuations in bacterial colonies," *Proceedings of the National Academy of Sciences*, vol. 107, no. 31, pp. 13 626–13 630, 2010.

[33] S. Saxena, F. Brémond, M. Thonnat, and R. Ma, "Crowd behavior recognition for video surveillance," in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2008, pp. 970–981.

[34] T. Ko, "A survey on behavior analysis in video surveillance for homeland security applications," in *Applied Imagery Pattern Recognition Workshop, 2008. AIPR'08. 37th IEEE*. IEEE, 2008, pp. 1–8.

[35] B. Zhou, X. Tang, and X. Wang, "Learning collective crowd behaviors with dynamic pedestrian-agents," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 50–68, 2015.

[36] L. Huang, T. Chen, Y. Wang, and H. Yuan, "Congestion detection of pedestrians using the velocity entropy: A case study of love parade 2010

disaster," *Physica A: Statistical Mechanics and its Applications*, vol. 440, pp. 200–209, 2015.

[37] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 18–32, 2014.

[38] R. Chaker, Z. Al Aghbari, and I. N. Junejo, "Social network model for crowd anomaly detection and localization," *Pattern Recognition*, vol. 61, pp. 266–281, 2017.

[39] Y. Benabbas, N. Ihaddadene, and C. Djeraba, "Motion pattern extraction and event detection for automatic visual surveillance," *EURASIP Journal on Image and Video Processing*, vol. 2011, no. 1, p. 163682, 2010.

[40] A. Abdelghany, K. Abdelghany, H. Mahmassani, and W. Alhalabi, "Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities," *European Journal of Operational Research*, vol. 237, no. 3, pp. 1105–1118, 2014.

[41] J. E. Almeida, R. J. Rosseti, and A. L. Coelho, "Crowd simulation modeling applied to emergency and evacuation simulations using multi-agent systems," *arXiv preprint arXiv:1303.4692*, 2013.

[42] W. K. Chow and C. M. Ng, "Waiting time in emergency evacuation of crowded public transport terminals," *Safety Science*, vol. 46, no. 5, pp. 844–857, 2008.

[43] J. D. Sime, "Crowd psychology and engineering," *Safety science*, vol. 21, no. 1, pp. 1–14, 1995.

[44] L. Lu, C.-Y. Chan, J. Wang, and W. Wang, "A study of pedestrian group behaviors in crowd evacuation based on an extended floor field cellular automaton model," *Transportation Research Part C: Emerging Technologies*, 2016.

[45] K. Al-Kodmany, "Crowd management and urban design: New scientific approaches," *Urban Design International*, vol. 18, no. 4, pp. 282–295, 2013.

[46] A. J. Lipton, P. L. Venetianer, N. Haering, P. C. Brewer, W. Yin, Z. Zhang, L. Yu, Y. Hu, G. W. Myers, A. J. Chosak *et al.*, "Video analytics for retail business process monitoring," Oct. 13 2015, uS Patent 9,158,975.

[47] M. C. Mongeon, R. P. Loce, and M. A. Shreve, "Busyness detection and notification method and system," Feb. 19 2015, uS Patent App. 14/625,960.

[48] E. A. Bernal, Q. Li, and R. P. Loce, "System and method for video-based detection of drive-offs and walk-offs in vehicular and pedestrian queues," May 16 2014, uS Patent App. 14/279,652.

[49] S. Gustafson, H. Arumugam, P. Kanyuk, and M. Lorenzen, "Mure: fast agent based crowd simulation for vfx and animation," in *ACM SIGGRAPH 2016 Talks*. ACM, 2016, p. 56.

[50] H. Perez, B. Hernandez, I. Rudomin, and E. Ayguade, "Task-based crowd simulation for heterogeneous architectures," in *Innovative Research and Applications in Next-Generation High Performance Computing*. IGI Global, 2016, pp. 194–219.

[51] J. C. Klontz and A. K. Jain, "A case study on unconstrained facial recognition using the boston marathon bombings suspects," *Michigan State University, Tech. Rep*, vol. 119, no. 120, p. 1, 2013.

[52] J. R. Barr, K. W. Bowyer, and P. J. Flynn, "The effectiveness of face detection algorithms in unconstrained crowd scenes," in *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*. IEEE, 2014, pp. 1020–1027.

[53] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–7.

[54] A. B. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 545–551.

[55] ——, "Counting people with low-level features and bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2012.

[56] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1215–1219.

[57] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2913–2920.

[58] K. Chen, S. Gong, T. Xiang, and C. Change Loy, "Cumulative attribute space for age and crowd density estimation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2467–2474.

[59] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 833–841.

[60] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3253–3261.

[61] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 640–644.

[62] K. Kang and X. Wang, "Fully convolutional neural networks for crowd segmentation," *arXiv preprint arXiv:1411.4464*, 2014.

[63] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami, "Fast crowd segmentation using shape indexing," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[64] S. Bandini, A. Gorrini, and G. Vizzari, "Towards an integrated approach to crowd analysis and crowd synthesis: A case study and first results," *Pattern Recognition Letters*, vol. 44, pp. 16–29, 2014.

[65] J. Shao, C. Change Loy, and X. Wang, "Scene-independent group profiling in crowd," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2219–2226.

[66] Z. Cheng, L. Qin, Q. Huang, S. Yan, and Q. Tian, "Recognizing human group action by layered model with multiple cues," *Neurocomputing*, vol. 136, pp. 124–135, 2014.

[67] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2871–2878.

[68] S. Yi, X. Wang, C. Lu, J. Jia, and H. Li, "L0 regularized stationary-time estimation for crowd analysis," *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[69] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, "Density-aware person detection and tracking in crowds," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2423–2430.

[70] F. Zhu, X. Wang, and N. Yu, "Crowd tracking with dynamic evolution of group structures," in *European Conference on Computer Vision*. Springer, 2014, pp. 139–154.

[71] J. Shao, K. Kang, C. C. Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 4657–4666.

[72] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes." in *CVPR*, vol. 249, 2010, p. 250.

[73] G. J. Brostow and R. Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 594–601.

[74] S. Chen, A. Fern, and S. Todorovic, "Person count localization in videos from noisy foreground and detections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1364–1372.

[75] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*. Springer, 2013, pp. 347–382.

[76] B. Xu and G. Qiu, "Crowd density estimation based on rich features and random projection forest," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–8.

[77] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[78] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[79] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[80] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 878–885.

[81] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 10, pp. 1713–1727, 2008.

[82] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[83] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[84] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1. IEEE, 2005, pp. 90–97.

[85] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[86] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[87] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[88] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[89] S.-F. Lin, J.-Y. Chen, and H.-X. Chao, "Estimation of number of people in crowded scenes using perspective transformation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 6, pp. 645–654, 2001.

[90] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247–266, 2007.

[91] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 7, pp. 1198–1211, 2008.

[92] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Digital Image Computing: Techniques and Applications, 2009. DICTA'09.* IEEE, 2009, pp. 81–88.

[93] N. Paragios and V. Ramesh, "A mrf-based approach for real-time subway monitoring," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–1034.

[94] A. Marana, L. d. F. Costa, R. Lotufo, and S. Velastin, "On the efficacy of texture analysis for crowd monitoring," in *Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI'98. International Symposium on*. IEEE, 1998, pp. 354–361.

[95] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, "Fast crowd density estimation with convolutional neural networks," *Engineering Applications of Artificial Intelligence*, vol. 43, pp. 81–88, 2015.

[96] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1299–1302.

[97] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun, "Crowd counting via weighted vlad on dense attribute feature maps," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[98] S. Kumagai, K. Hotta, and T. Kurita, "Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting," *arXiv preprint arXiv:1703.09393*, 2017.

[99] M. Marsden, K. McGuiness, S. Little, and N. E. O'Connor, "Fully convolutional crowd counting on highly congested scenes," *arXiv preprint arXiv:1612.00220*, 2016.

[100] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 785–800.

[101] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *European Conference on Computer Vision*. Springer, 2016, pp. 483–498.

[102] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 712–726.

[103] V. A. Sindagi and V. M. Patel, "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 IEEE International Conference on*. IEEE, 2017.

[104] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[105] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[106] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3642–3649.

[107] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *European Conference on Computer Vision*. Springer, 2018, pp. 757–773.

[108] D. Babu Sam, N. N. Sajjan, R. Venkatesh Babu, and M. Srinivasan, "Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3618–3626.

[109] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[110] Z. Shi, L. Zhang, Y. Liu, X. Cao, Y. Ye, M.-M. Cheng, and G. Zheng, "Crowd counting with deep negative correlation learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[111] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[112] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *European Conference on Computer Vision*. Springer, 2018, pp. 278–293.

[113] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding," *arXiv preprint arXiv:1811.11968*, 2018.

[114] J. Wan, W. Luo, B. Wu, A. B. Chan, and W. Liu, "Residual regression with semantic prior for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4036–4045.

[115] M. Zhao, J. Zhang, C. Zhang, and W. Zhang, "Leveraging heterogeneous auxiliary tasks to assist crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 736–12 745.

[116] V. Sindagi and V. Patel, "Inverse attention guided deep crowd counting network," *arXiv preprint*, 2019.

[117] V. A. Sindagi and V. M. Patel, "Ha-ccn: Hierarchical attention-based crowd counting network," *arXiv preprint arXiv:1907.10255*, 2019.

[118] ——, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1002–1012.

[119] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder network," *arXiv preprint arXiv:1903.00853*, 2019.

[120] M. Shi, Z. Yang, C. Xu, and Q. Chen, "Revisiting perspective information for efficient crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7279–7288.

[121] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5099–5108.

[122] Q. Zhang and A. B. Chan, "Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8297–8306.

[123] J. Wan and A. Chan, "Adaptive density map generation for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1130–1139.

[124] D. B. Sam, S. V. Peri, A. Kamath, R. V. Babu *et al.*, "Locate, size and count: Accurately resolving people in dense crowds via detection," *arXiv preprint arXiv:1906.07538*, 2019.

[125] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6142–6151.

[126] Q. Zhang and A. B. Chan, "3d crowd counting via multi-view fusion with 3d gaussian kernels," *arXiv preprint arXiv:2003.08162*, 2020.

[127] V. A. Sindagi and V. M. Patel, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *Pattern Recognition Letters*, 2017.

[128] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, "Cnn-based density estimation and crowd counting: A survey," *arXiv preprint arXiv:2003.12783*, 2020.

[129] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[130] Q. Wang, J. Gao, W. Lin, and X. Li, "Nwpu-crowd: A large-scale benchmark for crowd counting," *arXiv preprint arXiv:2001.03360*, 2020.

[131] A. Bansal and K. Venkatesh, "People counting in high density crowds from still images," *arXiv preprint arXiv:1507.08445*, 2015.

[132] J.-C. Chen, A. Kumar, R. Ranjan, V. M. Patel, A. Alavi, and R. Chellappa, "A cascaded convolutional neural network for age estimation of unconstrained faces," in *International Conference on BTAS*. IEEE, 2016, pp. 1–8.

[133] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *IEEE CVPR*, 2016, pp. 3150–3158.

[134] R. Ranjan, V. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE transactions on PAMI*, 2016.

[135] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*. Springer, 2014, pp. 346–361.

[136] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.

[137] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[138] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[139] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1265–1274.

[140] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[141] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[142] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[143] H. Zhang and K. Dana, "Multi-style generative network for real-time transfer," *arXiv preprint*, 2017.

[144] H. Zhang, V. A. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *arXiv preprint arXiv:1701.05957*, 2017.

[145] ——, "Joint transmission map estimation and dehazing using deep networks," *arXiv preprint arXiv:1701.05957*, 2017.

[146] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE TIP*, vol. 13, no. 4, pp. 600–612, 2004.

[147] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[148] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6298–6306.

[149] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *Proceedings of the IEEE*

Conference on Computer Vision and Pattern Recognition, 2018, pp. 1179–1188.

[150] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action recognition using visual attention," *arXiv preprint arXiv:1511.04119*, 2015.

[151] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[152] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3492–3506, 2017.

[153] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.

[154] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

[155] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 842–850.

[156] S. Chen, X. Tan, B. Wang, and X. Hu, "Reverse attention for salient object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 234–250.

[157] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European Conference on Computer Vision*. Springer, 2014, pp. 297–312.

[158] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[159] I. Loshchilov and F. Hutter, "Online batch selection for faster training of neural networks," *arXiv preprint arXiv:1511.06343*, 2015.

[160] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[161] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1091–1100.

[162] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection."

[163] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

[164] V. Sindagi and V. Patel, "Dafe-fd: Density aware feature enrichment for face detection," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 2185–2195.

[165] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," *arXiv preprint arXiv:1904.01649*, 2019.

[166] L. Wang, V. Sindagi, and V. Patel, "High-quality facial photo-sketch synthesis using multi-adversarial networks," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE, 2018, pp. 83–90.

[167] H. Zhang, V. Sindagi, and V. M. Patel, "Multi-scale single image dehazing using perceptual pyramid deep network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 902–911.

[168] H. Zheng, J. Fu, T. Mei, and J. Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition."

[169] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *European Conference on Computer Vision*. Springer, 2016, pp. 451–466.

[170] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[171] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[172] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2018.

[173] A. Roy and S. Todorovic, "A multi-scale cnn for affordance segmentation in rgb images," in *European conference on computer vision*. Springer, 2016, pp. 186–201.

[174] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis, "Ssh: Single stage headless face detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4875–4884.

[175] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. Torr, "Deeply supervised salient object detection with short connections," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3203–3212.

[176] V. Ramanishka, A. Das, J. Zhang, and K. Saenko, "Top-down visual saliency guided by captions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7206–7215.

[177] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, "Progressive attention guided recurrent network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 714–722.

[178] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 447–456.

[179] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[180] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 121–135, 2017.

[181] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[182] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[183] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 519–534.

[184] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*. Springer, 2016, pp. 75–91.

[185] A. Shrivastava, R. Sukthankar, J. Malik, and A. Gupta, "Beyond skip connections: Top-down modulation for object detection," *arXiv preprint arXiv:1612.06851*, 2016.

[186] F. Yang, X. Li, H. Cheng, Y. Guo, L. Chen, and J. Li, "Multi-scale bidirectional fcn for object skeleton extraction," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[187] W. Zhao, F. Zhao, D. Wang, and H. Lu, "Defocus blur detection via multi-stream bottom-top-bottom fully convolutional network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3080–3088.

[188] S. Z. Li, "Markov random field models in computer vision," in *European conference on computer vision*. Springer, 1994, pp. 361–370.

[189] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk *et al.*, "Slic superpixels," *Ecole Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, vol. 149300, pp. 155–162, 2010.

[190] S. Beucher *et al.*, "The watershed transformation applied to image segmentation," *SCANNING MICROSCOPY-SUPPLEMENT-*, pp. 299–299, 1992.

[191] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 2, 2017, p. 5.

[192] J. Kim, J. Kwon Lee, and K. Mu Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646–1654.

[193] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *The IEEE conference on computer vision and pattern recognition (CVPR) workshops*, vol. 1, no. 2, 2017, p. 4.

[194] W. Ke, J. Chen, J. Jiao, G. Zhao, and Q. Ye, "Srn: Side-output residual network for object symmetry detection in the wild," *arXiv preprint arXiv:1703.02243*, 2017.

[195] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[196] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2017, pp. 103–110.

[197] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[198] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[199] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1774–1783.

[200] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1713–1721.

[201] F. Saleh, M. S. A. Akbarian, M. Salzmann, L. Petersson, S. Gould, and J. M. Alvarez, "Built-in foreground/background prior for weakly-supervised semantic segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 413–432.

[202] A. Chaudhry, P. K. Dokania, and P. H. Torr, "Discovering class-specific pixels for weakly-supervised semantic segmentation," in *British Machine Vision Conference (BMVC)*, 2017.

[203] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Advances in neural information processing systems*, 1998, pp. 570–576.

[204] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[205] D. B. Sam, S. V. Peri, N. Mukuntha, and R. V. Babu, "Going beyond the regression paradigm with accurate dot prediction for dense crowds," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2020, pp. 2853–2861.

[206] C. Change Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2256–2263.

[207] R. Yasarla, V. A. Sindagi, and V. M. Patel, "Syn2real transfer learning for image deraining using gaussian processes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2726–2736.

[208] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2018, pp. 3235–3246.

[209] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *arXiv preprint arXiv:1610.02242*, 2016.

[210] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[211] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks."

[212] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[213] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[214] C. Zhang, K. Kang, H. Li, X. Wang, R. Xie, and X. Yang, "Data-driven crowd understanding: A baseline for a large-scale crowd dataset," *IEEE Transactions on Multimedia*, vol. 18, no. 6, pp. 1048–1061, 2016.

[215] V. A. Sindagi, R. Yasarla, and V. M. Patel, "Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1221–1231.

[216] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685–1692.

[217] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 532–539.

[218] B. Wu, S. Lyu, B.-G. Hu, and Q. Ji, "Simultaneous clustering and tracklet linking for multi-face tracking in videos," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2856–2863.

[219] Y.-I. Tian, T. Kanade, and J. F. Cohn, "Recognizing action units for facial expression analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 97–115, 2001.

[220] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[221] X. Di, V. A. Sindagi, and V. M. Patel, "Gp-gan: gender preserving gan for synthesizing faces from landmarks," *arXiv preprint arXiv:1710.00962*, 2017.

[222] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 1, pp. 39–51, 1998.

[223] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2879–2886.

[224] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *European Conference on Computer Vision*. Springer, 2014, pp. 720–735.

[225] Y. Liu, H. Li, J. Yan, F. Wei, X. Wang, and X. Tang, "Recurrent scale approximation for object detection in cnn," in *IEEE international conference on computer vision*, vol. 5, 2017.

[226] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille, "Single-shot object detection with enriched semantics," Center for Brains, Minds and Machines (CBMM), Tech. Rep., 2018.

[227] W. Ren, D. Kang, Y. Tang, and A. B. Chan, "Fusing crowd density maps and visual object trackers for people tracking in crowd scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5353–5362.

[228] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988.

[229] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2359–2367.

[230] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," in *Deep Learning for Biometrics*. Springer, 2017, pp. 57–79.

[231] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 390–399.

[232] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *IEEE CVPR*, 2017.

[233] Y.-L. Hou and G. K. Pang, "People counting and human detection in a challenging situation," *IEEE transactions on systems, man, and cybernetics-part a: systems and humans*, vol. 41, no. 1, pp. 24–33, 2011.

[234] D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 1187–1190.

[235] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," in *ICLR*, 2016.

[236] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[237] P. Samangouei, M. Najibi, L. Davis, and R. Chellappa, "Face-magnet: Magnifying feature maps to detect small faces," *arXiv preprint arXiv:1803.05258*, 2018.

[238] E. Ohn-Bar and M. M. Trivedi, "To boost or not to boost? on the limits of boosted trees for object detection," in *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2016, pp. 3350–3355.

[239] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[240] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3676–3684.

[241] A. G. Howard, "Some improvements on deep convolutional neural network based image classification," *arXiv preprint arXiv:1312.5402*, 2013.

[242] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 516–520.

[243] R. Ranjan, V. M. Patel, and R. Chellappa, "A deep pyramid deformable part model for face detection," in *Biometrics Theory, Applications and Systems (BTAS), 2015 IEEE 7th International Conference on*. IEEE, 2015, pp. 1–8.

[244] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.